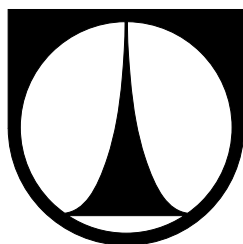


TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií



Řízení pohonu mikrotitrační destičky a odběrové jehly v průtokovém cytometru

Bakalářská práce



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B 2612 – Elektrotechnika a informatika

Obor: 2612R011 – Elektronické informační a řídicí systémy

**Řízení pohonu mikrotitrační destičky a
odběrové jehly v průtokovém cytometru**

**Control of a microtiter plate and sample
syringe mechanical positioning system in a
flow cytometry device**

Bakalářská práce

Autor práce: **Jiří Kuba**

Vedoucí práce: doc. Ing. Ivan Doležal, CSc.

Konzultant práce: MUDr. Jan Wolf



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jiří Kuba**
Osobní číslo: **M09000360**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Řízení pohonu mikrotitrační destičky a odběrové jehly v průtokovém cytometru**
Zadávací katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se u konzultanta s účelem a koncepcí polohovaného mechanismu a s požadavky na pohon a jeho ovládání.
2. Seznamte se s parametry KM.
3. Seznamte se s principy řízení KM vč. PWM a mikrokrokování, kompenzací, detekce zablokování a obnovy polohy, řízení po rampách.
4. Proveďte rešerši jednodeskových řídicích modulů a integrovaných řadičů/budičů KM.
5. Vyberte vhodné řešení pro řízení KM.
6. Vyberte vhodný mikrořadič a příp. adaptérový obvod pro ovládání řídicích obvodů z PC po USB.
7. Navrhněte, osadte a oživte desku plošných spojů s vybranými obvody.
8. Navrhněte a odlaďte firmware pro řízení pohonu, navrhněte vhodný soubor příkazů pro řízení z PC.
9. Vyzkoušejte vaše řešení na prototypu mechanismu.
10. Vytvořte demonstrační aplikaci na PC pro odzkoušení řízení pohonu.



Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30–40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] Rydlo, P.: Krokové motory a jejich řízení. [Skriptum FM] TUL, Liberec 2000.
- [2] Jones D W: Control of Stepping Motors. A Tutorial. The University of Iowa. <http://www.divms.uiowa.edu/jones/step>
- [3] Zhu J: Electromechanical Systems. Stepping Motors and Their Power Electronic Drives. University of Technology Sydney.
http://services.eng.uts.edu.au/cempe/subjects_JGZ/ems/ems_ch10_nt.pdf
- [4] Kniha o programování vybrané řady mikrořadičů v jazyce C, např. Váňa, V.: Atmel AVR programování v jazyce C. BEN, Praha 2003. ISBN 80-7300-102-0.
- [5] Datové listy vybraných řadičů/budičů KM na webu výrobce.
- [6] Datový list použitého mikrořadiče na webu výrobce.

Vedoucí bakalářské práce: doc. Ing. Ivan Doležal, CSc.

Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce: MUDr. Jan Wolf

Wolf & Danniell s.r.o., Praha

Datum zadání bakalářské práce: 14. října 2011

Termín odevzdání bakalářské práce: 18. května 2012

prof. Ing. Václav Kopecký, CSc.

děkan



doc. Ing. Petr Tůma, CSc.

vedoucí ústavu



Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis



Poděkování

Děkuji doc. Ing. Ivanu Doležalovi, CSc. za poskytnutí cenných rad a odborného vedení při zpracování této bakalářské práce. Také děkuji firmě W&D za pomoc při shánění součástek a finanční podporu vývoji modulu. Nakonec bych chtěl poděkovat mým rodičům za trpělivost a podporu po celou dobu studia.



Abstrakt

Tato bakalářská práce se zabývá vývojem modulu pro pohon mikrotitrační destičky a odběrové jehly v průtokovém cytometru. Základem modulu jsou krokové motory a kontroléry STM L6470, které jsou relativně novým prvkem na trhu. Kontroléry zajišťují jak buzení vinutí motorů, tak provádění příkazů pro pohyb motorů. Také je popsáno jejich zapojení a řízení. Deska je navrhována v programu Eagle, knihovna funkcí pro ovládání modulu je napsaná v jazyce C#.

Klíčová slova

Krokový motor, kontrolér, řízení, průtokový cytometr, pohon, STM L6470.

Abstract

This bachelor thesis is focused on the research of module for driving and controlling of a microtiter plate and the sample syringe mechanical positioning system in a flow cytometry device. Base of the module are stepper motors and controllers STM L6470, which are relatively new on the market. Controllers provide excitation of motor windings and execution of driving commands. Printed circuit board is designed in the Eagle, dynamic linked library with driving commands for module is programmed in C#.

Key words

Stepper motor, controller, drive, flow cytometry device, STM L6470.



Obsah

Zadání bakalářské práce	3
Prohlášení	5
Poděkování	6
Abstrakt	7
Klíčová slova	7
Abstract	7
Key words	7
Obsah	8
Seznam obrázků	11
Seznam tabulek	11
1 Úvod	12
2 Krokový motor	13
2.1 Princip činnosti a dělení krokových motoru	13
2.2 Parametry krokových motorů	14
2.3 Výběr krokového motoru	15
2.4 Buzení krokových motorů	16
2.5 Kompenzace zpětného elektromotorického napětí	17
2.6 Detekce zablokování	17
3 Rozbor návrhu řídicího modulu	19
3.1 Výkonová část	19
3.2 Výběr kontroléru	20
3.3 Komunikační část	22
3.4 Nástin řešení	22



4	Popis STM L6470	23
4.1	Komunikace po SPI	23
4.2	Provoz L6470	24
4.3	Příkazy pro L6470	24
4.4	Registry L6470	24
4.5	Kompenzace	26
4.6	Stručný průvodce pro zprovoznění L6470	28
5	Schéma zapojení modulu	30
5.1	Napájení modulu	30
5.2	Zapojení L6470	31
5.3	Napojení na sběrnici SPI	32
5.4	Deska plošných spojů	32
5.4.1	Konstrukce DPS	32
5.4.2	Popis DPS	33
5.4.3	Redukce pro SPI - USB konvertor s FT2232	35
6	Knihovna pro ovládání řídicího modulu	37
6.1	Příklad použití DLL	39
7	Demonstrační aplikace	40
	Závěr	43
	Literatura	45
	Přílohy	46



Seznam obrázků

1	Řez hybridním krokovým motorem. [3]	14
2	Statický moment KM.	15
3	Momentová charakteristika KM. [2]	15
4	Mikrokrokování 1/128. [6]	17
5	Posun fáze elektromotorického napětí. [8]	18
6	Schématické znázornění připojení modulu k nadřazeným systémům.	22
7	Blokové schéma L6470. [6]	23
8	Kompenzace BEMF. [6]	26
9	Kompenzace poklesu V_S . [6]	27
10	Schéma zapojení stabilizátoru.	30
11	Schema zapojení L6470.	31
12	Schéma napojení modulu na SPI.	33
13	Detail provedení chlazení kontroléru.	34
14	Celkový pohled na DPS redukce.	36
15	Pohled na horní stranu spojů.	36
16	Pohled na spodní stranu spojů.	36
17	Rozmístění součástek redukce.	36
18	Diagram knihovny pro ovládání modulu.	38
19	Nastavení parametrů motoru.	40
20	Hlavní okno demonstrační aplikace.	41
21	Registry kontroléru.	42
22	Demonstrace pohonu odběrové jehly.	42
23	Celkový pohled na DPS modulu.	46
24	Pohled na horní stranu spojů DPS modulu.	47
25	Pohled na spodní stranu spojů DPS modulu (zrcadleno).	47
26	Pohled na stranu součástek DPS modulu.	48
27	Úplné schéma zapojení modulu.	49
28	Fotka celku DPS.	50
29	Fotka detailu DPS.	50



30	Konvertor z USB na SPI.	51
31	Redukce pro konvertor z USB na SPI.	51

Seznam tabulek

1	Třída KontrolerL6470.	52
2	Interface IUSBinterface.	53
3	Třída KontrolerL6470 pokračování.	54
4	Třída Pohon.	54
5	Seznam součástí.	55



1 Úvod

Cílem této práce je návrh modulu pro řízení pohonu mikrotitrační destičky a odběrové jehly v průtokovém cytometru. Předpokládá se použití simultánního trojosého řízení pomocí krokových motorů. K modulu jsem naprogramoval knihovnu funkcí pro jeho snadné ovládání a demonstrační aplikaci.

Téma této práce navrhla firma W&D, která se zabývá vývojem, prodejem a servisem laboratorních přístrojů.

Průtokový cytometr je přístroj, který pomocí laseru určuje vlastnosti a složení drobných částic ve formě suspenze. Vzorky částic se do cytometru obvykle vkládají umístěné na destičce tvořené čtvercovou sítí zkumavek. Ze zkumavek jsou vzorky odebírány odběrovou jehlou. Obvykle se analyzuje velké množství vzorků najednou a analýza jednoho vzorku trvá jen krátký okamžik. Je proto třeba zajistit dostatečně rychlý přísun vzorků do přístroje.

Tato práce mě zaujala, protože bude zakončena praktickou realizací modulu. Také není úzce zaměřená na jedno téma, zahrnuje jak návrh a výrobu desky plošných spojů, tak programování knihovny funkcí pro ovládání modulu.



2 Krokový motor

V této kapitole je stručně popsán krokový motor a jeho vlastnosti se zaměřením na hybridní krokové motory. Podrobnější informace lze najít například v publikacích [1] a [2].

Krokový motor se liší od ostatních typů elektrických motorů tím, že jeho pohyb je nespojitý. Pohybuje se po částech – krocích, a proto jsme schopni přesně nastavit jeho pozici a udržet ji i po zastavení motoru. Má širokou škálu uplatnění. Používá se všude tam, kde se klade důraz na přesné nastavení polohy, např. v robotice, průmyslových aplikacích, tiskárnách, výpočetní technice, CNC systémech apod.

Mezi jeho výhody patří nízká poruchovost díky jednoduché konstrukci, přesné určení polohy a otáček hřídele bez externích prvků. Mezi jeho hlavní nevýhody patří nižší poměr výkonu a hmotnosti, trvalý odběr proudu v klidovém stavu a pokles momentu se zvyšující se rychlostí otáček. Mohou se vyskytnout lokální poklesy momentu v souvislosti s rezonanční frekvencí motoru. Také hrozí ztráta údaje o poloze nazývaná ztráta kroku. Ta nastává při zatížení motoru větším, než je maximální přípustný moment pro danou rychlost.

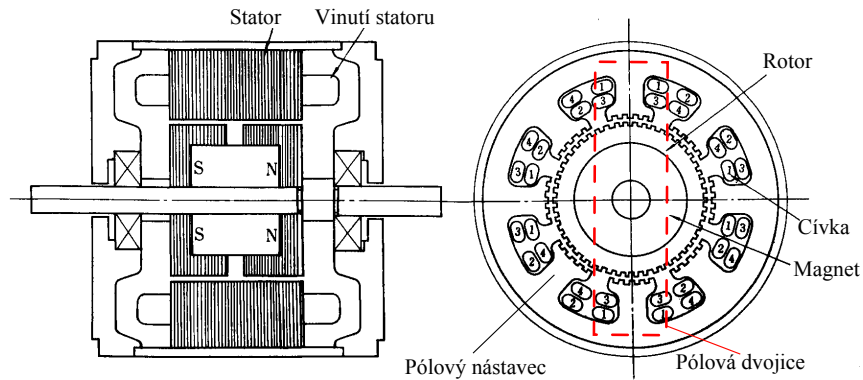
2.1 Princip činnosti a dělení krokových motorů

Stator krokového motoru je tvořen rotorem a několika cívkami, které tvoří fáze motoru. Fáze jsou navinuty na pólových nástavcích tvořících protilehlé dvojice, viz obrázek 1. Sekvenčním spínáním cívek se vytváří magnetické pole, které působí na rotor a zajišťuje jeho pohyb po jednotlivých krocích.

Krokové motory se dělí na 3 základní skupiny podle **typu rotoru**:

- Motor s aktivním rotorem
- Motor s proměnnou reluktancí
- Hybridní motor

Hybridní krokové motory jsou v současnosti nejrozšířenějším typem krokových motorů. Tvoří je kombinace motoru s aktivním rotorem a motoru s proměnnou reluktancí. Jejich spojení zvyšuje počet kroků na otáčku a zlepšuje dynamické vlastnosti KM (krokového motoru) při stejné hmotnosti. Typický počet kroků na otáčku je 200. Stator je tvořen pólovými



Obrázek 1: Řez hybridním krokovým motorem. [3]

nástavci s navinutými cívkami, viz obrázek 1.

Rotor hybridního motoru je tvořen magneticky měkkou ocelí a uvnitř má axiálně polarizované permanentní magnety. Po obvodu rotoru jsou drážky rovnoběžné s osou rotace, takzvané zuby. Jejich rozteč je shodná s roztečí zubů na pólových nástavcích. Rotor je rozdělen na dvě sekce, které mají zuby vůči sobě posunuté o polovinu jejich rozteče.

Jak od sebe rozeznat jednotlivé druhy KM? Motor s aktivním rotorem klade při roztočení hřídele prsty mírný odpor při průchodu magnetu kolem pólů. Motor s proměnnou reluktancí se točí volně, mírný odpor může vyvíjet zbytkové zmagnetování rotoru. Hybridní motory jsou z pohledu uživatele nerozlišitelné od motorů s aktivním rotorem.

2.2 Parametry krokových motorů

V této kapitole se pracuje s termíny uvedenými v publikaci [1]. Uvádím jen nejdůležitější parametry KM.

Délka kroku je ovlivněna počtem fází a počtem zubů na rotoru. Vypočítá se ze vztahu (1), kde Φ je úhel ve stupních, N je počet fází a m je počet zubů rotoru.

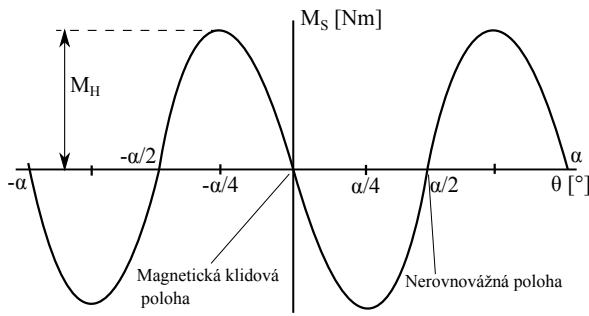
$$\Phi = \frac{360}{Nm} [^\circ] \quad (1)$$

Jmenovitý přídržný moment M_H [Nm] je přídržný moment KM s buzenou jednou cívkou jmenovitým proudem.

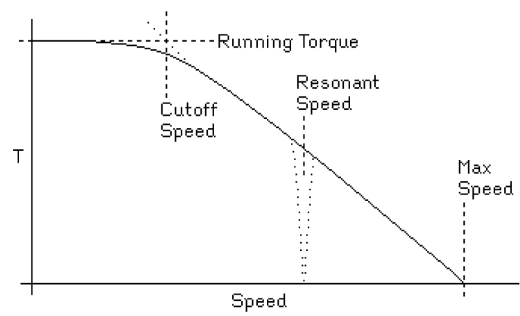
Statický moment M_S [Nm] je moment potřebný k udržení motoru v klidu při zatížení momentem M_A [Nm]. Dojde k vychýlení rotoru o úhel θ [rad]. Pro ideální dvoufázový hybridní motor platí rovnice (2). Ve výpočtu se uplatní jmenovitý přídržný moment M_H a délka kroku α [rad]. Průběh je znázorněn na obrázku 2.

$$\begin{aligned} M_S &= -M_H \sin\left(\frac{2\pi\theta}{\alpha}\right) \\ 0 &= M_A + M_S \end{aligned} \quad (2)$$

Dynamický moment krokového motoru úzce souvisí s momentovou charakteristikou KM, viz obrázek 3. Na obrázku je patrný pokles krouticího momentu v důsledku rezonancí KM.



Obrázek 2: Statický moment KM.



Obrázek 3: Momentová charakteristika KM. [2]

2.3 Výběr krokového motoru

Pro tuto kapitolu je jako výchozí zdroj použit materiál [9]. KM je třeba vybírat s ohledem na požadavky pro danou aplikaci. Klíčové je stanovení hnacího momentu M_H . Nejprve je třeba určit časy pro dosažení požadované polohy, maximální rychlost a startovací rychlost. Z toho se spočítají hodnoty zrychlení a_0 mezi startovací a maximální rychlostí. Dále je třeba zjistit setrvačný moment J_s poháněného mechanismu a zatěžovací moment M_z vyvolaný ostatními silami v mechanismu. Kritéria předběžného výběru:

- Setrvačný moment rotoru J_R nesmí být menší než $1/10$ setrvačného momentu zátěže J_S .
- Startovací krouticí moment při startovací rychlosti nesmí být menší než zatěžovací momenty M_z .



- krouticí moment motoru při maximální rychlosti není menší než zatěžovací moment M_z .

Dále je třeba vypočítat krouticí moment při zrychlení a_0 :

$$M_a = (J_s + J_R) \cdot a_0$$

Na závěr se spočítá výsledný hnací moment M_H , kde x je bezpečnostní faktor zvolený v rozmezí 1,5 až 2:

$$M_H = (M_a + M_z) \cdot x$$

Také je třeba zohlednit prostředí, v němž bude motor pracovat a nároky na jeho chlazení.

2.4 Buzení krokových motorů

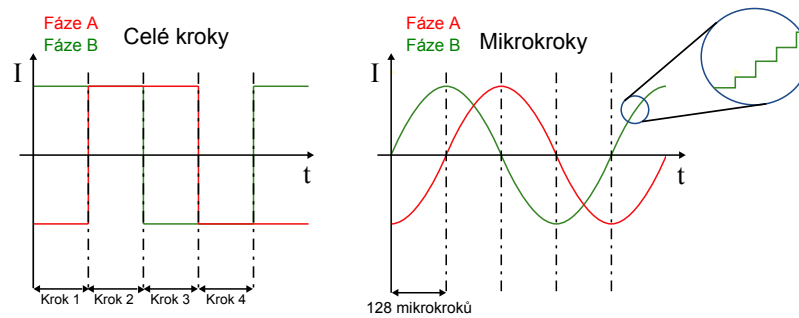
Dříve převládalo **unipolární buzení**, protože konstrukce budících prvků byla jednodušší a levnější. Na jednu fázi připadá jeden tranzistor. Proud prochází cívkou vždy jedním směrem a je buzena vždy jen jedna cívka. Takto se využije pouze 25% výkonu motoru.

Druhý způsob je **bipolární buzení**. Proud může procházet cívkou oběma směry. To umožňuje vybudit cívky i v opačném směru, čímž stoupne efektivita motoru. Bipolární buzení vyžaduje čtyři tranzistory na fázi. S rozmachem integrovaných obvodů se smazal rozdíl v ceně mezi unipolárním a bipolárním buzením. Proto se dnes pro motory s proměnnou reluktancí a pro hybridní motory používá bipolární buzení.

Pro regulaci napětí na cívkách se u modernějších budičů používá metody pulsně šířkové modulace (PWM). Spočívá v přivedení plného napětí v pulsech s proměnnou střídou. Tím se minimalizují ztráty na tranzistorech, které je mají nejmenší v plně otevřeném a zavřeném stavu.

Pokud je možné efektivně regulovat napětí na cívce, může se použít metoda mikrokrokování. Mikrokrokování se používá pro natočení KM do mezipoloh mezi kroky. Využívá se proměnné síly přitahu cívky podle velikosti procházejícího proudu.

Proud se pouští do sousedních fází v určitých poměrech a podle toho se rotor ustálí v mezipoloze mezi dvěma kroky. Princip je znázorněn na obrázku 4.



Obrázek 4: Mikrokrokování 1/128. [6]

2.5 Kompenzace zpětného elektromotorického napětí

Při pohybu rotoru se natáčí i magnetické pole permanentních magnetů. Tím vzniká na cívkách statoru zpětné elektromotorické napětí, které snižuje hodnotu budicího napětí, tím pádem i proud potřebný k vytvoření magnetického pole cívek. To způsobuje snížení krouticího momentu KM a jeho maximální rychlosti. Tento jev nabývá na síle se zvyšující se rychlostí otáčení KM.

Jeden ze způsobů, jak tomuto jevu zabránit, je zvýšit budicí napětí na cívkách KM. Zvýšení napětí může vést k přehřívání motoru díky odporu vinutí cívek. Poté hrozí porušení izolace a zkrat, nebo demagnetizace permanentních magnetů rotoru.

Proto se vyrábějí kontroléry používající k buzení cívek vyšší napětí v kombinaci s proudovým omezením. Další z možností je měnit budicí napětí cívek podle otáček motoru.

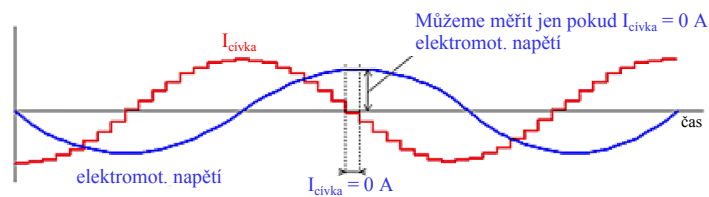
2.6 Detekce zablokování

Pokud nechceme k detekci zablokování KM využít externí komponenty, jako jsou otáčkoměry, můžeme použít metodu popsanou ve článku [8]. Metoda spočívá v měření proudů a elektromotorických napětí ve fázích KM.

Pokud KM běží nezatížený, elektromotorické napětí je nejvyšší, když je proud tekoucí cívkou nulový. V tento okamžik je vhodné elektromotorické napětí měřit. S narůstající zátěží se elektromotorické napětí posouvá do fáze s proudem, až zmizí v okamžiku zablokování KM, viz obrázek 5. Průvodním jevem je zvýšení rychlosti nárůstu proudu vinutím. Měřením těchto



parametrů se dá usuzovat na zablokování motoru.



Obrázek 5: Posun fáze elektromotorického napětí. [8]

Nevýhodou této metody je posuv fáze elektromotorického napětí při provozu KM na hranici zablokování. V tento okamžik je náročné elektromotorické napětí změřit. Další komplikace plyne z mechanických vlastností motoru. Zablokovaný motor může začít vibrovat a tím vytvářet malé elektromotorické napětí. Kontrolér poté může chybně vyhodnotit stav motoru. Proto se KM většinou nepoužívají na hraně provozních možností.



3 Rozbor návrhu řídicího modulu

Polohovací mechanismus průtokového cytometru má zajišťovat pohyb desky se vzorky ve dvou vodorovných osách X a Y. Ve třetí vertikální ose Z se bude pohybovat odběrová jehla.

Všechny osy musí mít dostatečnou dynamiku pohybu. Pohyb v osách X a Y musí dosahovat maximální rychlosti $20 \text{ m}\cdot\text{s}^{-1}$ a zrychlení $10 \text{ m}\cdot\text{s}^{-2}$. Osy budou pohybovat břemenem do hmotnosti 1 kg. Osa Z má stejné parametry jako osy horizontální, nesené břemeno bude mít hmotnost do 0,5 kg. Také je třeba počítat s pohybem a udržením polohy proti zemské přitažlivosti. Délka os bude v rozmezí do jednoho metru. V polohovacím mechanismu budou použity dvoufázové krokové motorky s nižším statickým momentem a jmenovitým proudem pod 2 A.

Požadavky na řídicí modul

Ovládání polohovacího mechanismu má zajišťovat řídicí modul. Podle podrobnějších informací získaných od konzultanta, jsem se rozhodl pro co nejuniverzálnější formu řešení. Modul by měl být schopný pracovat samostatně mimo přístroj, ideálně s řízením přes USB port počítače. Zároveň by mělo být možno zabudovat systém do cytometru bez větších úprav a ovládat ho mikroprocesorem, který bude řídit i ostatní funkce cytometru. Cena modulu musí odpovídat jeho funkcím a modul musí být vyrobitelný z dostupných součástek ve větším počtu kusů.

Možné způsoby konstrukce a výběr vhodného řešení

Konstrukce modulu je rozdělena na část výkonovou a logickou. Výkonová část bude zajišťovat přenos elektrické energie na cívky krokových motorů, část logická bude zajišťovat ovládání části výkonové a přenášet informace mezi modulem a nadřazeným systémem. Informace budou přenášeny po sériové lince nebo podobném jednodušším komunikačním protokolu. V případě ovládání modulu přímo z PC se k modulu připojí převodník na sběrnici USB.

3.1 Výkonová část

Pro konstrukci výkonové části se nabízejí tyto možnosti:



1. Sestavit šest H-můstků pro buzení třech KM, doplnit ochranné obvody a logiku pro jejich spínání, použít řízení mikroprocesorem.
2. Použít budič v podobě integrovaného obvodu s vestavěnými H-můstky a logikou pro jejich spínání, použít řízení mikroprocesorem.
3. Vybrat kontrolér s pokročilejšími funkcemi pro ovládání KM včetně logiky pro jejich řízení a komunikačního rozhraní.

Variantu číslo 1 je nevhodná kvůli složitosti konstrukce, nárokům na velikost DPS a dlouhým výrobním časům kvůli počtu součástek. Také zahrnutí pokročilých funkcí, jako je PWM, detekce ztráty kroku, zkratu, nízkého napětí apod., by návrh velmi zkomplikovalo.

Varianta číslo 2 těmito problémy trpí méně, je tu ale potíže s ovládáním budičů. Ty jsou řízeny pomocí impulsů. S každým impulsem se provede krok, nebo jeho část. Směr a způsob krokování se nastavuje pomocí dalších signálových pinů. Takové řešení je velmi oblíbené díky snadnému napojení na počítač přes paralelní port.

V současné době je tento způsob zastaralý. Paralelní port byl nahrazen portem USB a z počítačů postupně mizí. Způsob ovládání přes paralelní port by také vyžadoval neustálé generování pulsů ze strany počítače, nebo jiného nadřazeného systému a zbytečně ho zatěžoval. V případě zkolabování počítače by hrozilo zničení polohovacího mechanismu, např. při najetí za krajní polohu. Tyto nedostatky by se daly vyřešit použitím mikrořadiče, který se bude starat o generování impulsů, nastavení budičů a o tlumočení povelů z nadřazeného systému. Jenže v cytometru by se poté nacházely mikrořadiče dva, což není úplně elegantní řešení a navíc by poté modul nesplňoval požadavek na jednoduchost a nízkou cenu.

Proto byla zvolena varianta číslo 3, tedy zvolit vhodný IO s pokročilejšími funkcemi, který nebude vyžadovat neustálé posílání řídicích impulsů z nadřazeného systému, nebo přítomnost mikroprocesoru. Tím se zjednoduší konstrukce modulu, zvýší spolehlivost a sníží náklady na stavbu modulu.

3.2 Výběr kontroléru

Zaměřil jsem se na bipolární kontroléry určené pro dvoufázové KM schopné pracovat s napětím kolem 24 V a konstantním proudem do 2 A na fázi. Kontroléry musí mít integrované



H-můstky a sadu instrukcí pro jejich řízení. Musí být ovládané přes jednu ze standardizovaných sběrnic, jako je I²C, SPI, UART a pod. Také byla zohledněna cena a dostupnost součástky v ČR. Do užšího výběru byly zařazený tyto kontroléry:

Allegro A4980

Kontrolér Allegro A4980 [4] poskytuje proud ve špičce 1,4 A při 28 V. Má detekci zablokování KM, tepelnou pojistku, detekci zkratu, regulaci napětí pomocí PWM. Jeden krok KM dokáže rozdělit na 16 mikrokroků. Neobsahuje sadu instrukcí pro řízení KM, je vyžadován mikrokontrolér. Cena v době odevzdání práce byla 160,- Kč v maloobchodním prodeji.

Trinamic TMC260

Trinamic TMC260 [5] nabízí stejné funkce jako Allegro, jeden krok dokáže rozložit na 256 mikrokroků. Na cívku KM dokáže dodat 1,7 A při 40 V. Výrobce nabízí inteligentní regulaci proudu podle zátěžového momentu za účelem úspory energie a omezení přehřívání. Tato funkce by nebyla využita vzhledem k tomu, že 2 osy budou ve vodorovné poloze a nebude na ně působit žádná síla. Kontrolér neobsahuje sadu instrukcí pro řízení KM, nelze provozovat bez mikrokontroléru. Kontrolér navíc v době odevzdání práce stál 360,- Kč. To je příliš mnoho vzhledem k tomu, že budou použity tři kusy.

STM L6470

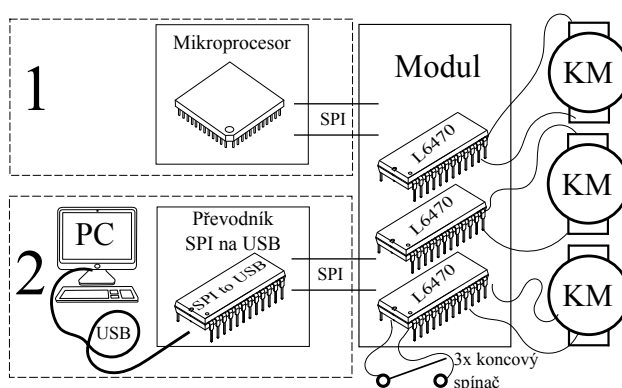
Nejlvýhodněji z výběru vyšel kontrolér L6470 [6] od STMicroelectronics. Je to jediný z kontrolérů, který dokáže splnit požadavky kladené na modul. Rozsah napájecího napětí je 8–45 V a dodávaný proud je 3 A r.m.s., špičkově až 7 A. Krok umí rozdělit na 128 mikrokroků, komunikuje po SPI a má bohatou sadu funkcí pro ovládání KM včetně PWM a různých kompenzací. Cena se pohybuje kolem \$4,5 a STM nabízí vzorky zdarma k vyzkoušení, čehož bylo využito.

Samozřejmě lze pořídit komplexní moduly pro řízení KM i se softwarem, cena se ale pohybuje v řádech tisíců až desetitisíců Kč. Pro použití v cytometru nejsou vhodné.



3.3 Komunikační část

Díky volbě kontroléru se zabudovanou logikou řízení se komunikační část modulu podstatně zjednodušila. Stačilo zvolit vhodný převodník z rozhraní SPI na sběrnici USB. Díky předchozí zkušenosti s produkty od FTDI z bakalářského projektu padla volba na přípravek s FTDI FT2232 [7]. Také byl vybrán proto, že ho škola měla nakoupený předem. Dále mě zaujal převodník od FTDI s FT232. Je to USB kabel s převodníkem zabudovaným v koncovce USB konektoru. Druhý konec kabelu tvoří desetipinový konektor, takže narozdíl od FT2232 není třeba použít redukci s napojením na konektor.



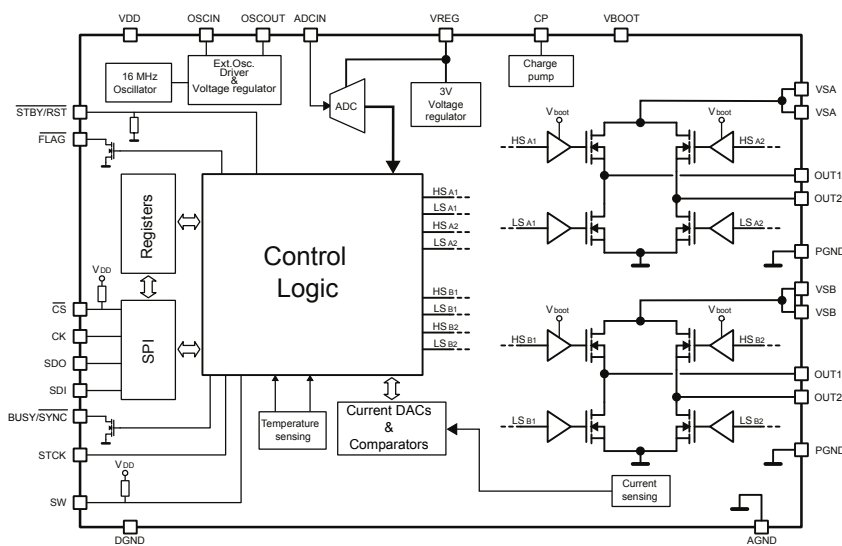
Obrázek 6: Schématické znázornění připojení modulu k nadřazeným systémům.

3.4 Nástin řešení

Po dokončení výběru hlavních součástí bylo jasné, jak bude vypadat hrubé schéma modulu. Na obrázku 6 je znázorněno připojení modulu k nadřazeným systémům. V ostrém provozu bude použita varianta číslo 1, kde bude modul řízen mikroprocesorem. V průběhu vývoje desky byla použita varianta číslo 2, modul byl připojen k počítači přes USB.

4 Popis STM L6470

V této kapitole jsou popsány důležité vlastnosti kontroléru L6470, který tvoří hlavní část modulu. Cílem není podrobně popsat vlastnosti kontroléru, ale spíše nastínit princip fungování a důležité funkce. Pro podrobné informace doporučuji prostudovat datasheet [6]. Základní údaje jsou k nalezení v kapitole 3.2. L6470 obsahuje dva H-můstky tvořené z tranzistorů typu DMOS, viz obrázek 7, na které se dá připojit jeden doufázový KM. Buzení KM je bipolární.



Obrázek 7: Blokové schéma L6470. [6]

4.1 Komunikace po SPI

Komunikace s L6470 probíhá po sběrnici SPI, kontrolér vystupuje jako slave. Max. rychlost je 5 Mbit/s. Data jsou do kontroléru přijímána po osmibitových slovech, MSB (bit s největší vahou) na prvním místě. Mezi slovy je třeba na nejméně 57 ns deaktivovat CS. Kontrolér data odesílá vždy o jedno slovo pozadu proti příkazu pro odeslání dat. Zároveň odesílá jen tehdy, pokud do něj data zapisujeme. Pokud kontrolér odesílá data a během přenosu přijme další příkaz na odeslání dat, přenos se přeruší a začnou se odesílat nová data.



4.2 Provoz L6470

Po připojení napájení se kontrolér nachází ve stavu reset. Tento stav neopustí, dokud nemá aktivní regulátor napětí, napájení není vyšší než 8 V a nezačne pracovat interní oscilátor. Poté přejde do provozního režimu. Výstupy motorů jsou ve stavu vysoké impedance. Tento stav opustí po přijetí jakéhokoliv pohybového příkazu.

4.3 Příkazy pro L6470

L6470 nedokáže pracovat samostatně, vždy je třeba mu posílat příkazy. Po zaslání příkazu proběhne vyhodnocení a příkaz se vykoná, nebo L6470 signalizuje nesplnitelný či chybný příkaz. Signalizace probíhá uzemněním pinu FLAG a navrácením kódu s chybným, nebo nesplnitelným příkazem po SPI. Po dobu vykonávání zasláného příkazu je aktivní pin BUSY. Tato konfigurace umožňuje rychlejší reakci nadřazeného systému, protože odpadá neustálé načítání stavového registru L6470. Piny FLAG a BUSY se dají zapojit dohromady pro více kontrolérů. Po odpojení napájení se registry L6470 resetují do defaultního nastavení.

Seznam příkazů, které kontrolér umí vykonat, se nachází v datasheetu v kapitole 9.2. Příkaz se skládá z prvního bytu, který ho identifikuje. Za prvním bytem následují další byty dle charakteru příkazu. Naříklad pokud je třeba roztočit motor danou rychlostí, vyšle se sekvence 01010001. Poslední bit určuje směr otáčení. Za ním musí následovat tři byty korespondující s registrem SPEED. Obdobně fungují všechny příkazy, které zapisují data do registru kontroléru.

Další z důležitých příkazů je příkaz pro přečtení hodnoty registru. Začíná sekvencí 001, za kterou následuje adresa registru. Poté je nutné vyslat tolik bytů, jako je délka načítaného registru zaokrouhlená na celé byty. Pokud chceme během načítání data pouze přijímat, stačí vysílat byty s hodnotou 0, v podstatě tedy stačí posílat hodinové pulsy po CLK. Pokud je do L6470 vyslán příkaz před dokončením předchozího příkazu, začne se provádět příkaz nový, nebo L6470 oznámí neproveditelný příkaz.

4.4 Registry L6470

Kontrolér L6470 obsahuje celou řadu registrů, které ovlivňují, nebo indikují jeho stav. Seznam všech registrů je v datasheetu v kapitole 9.1. Každý registr má svou adresu, pevnou délku a



přednastavenou hodnotu, která se obnoví při restartu kontroléru. Neméně důležitý je příznak, který určuje, kdy je z registru povoleno číst a kdy se do něj může zapisovat. Příznaky jsou R - vždy čitelný, WR - vždy zapisovatelný, WS - zápis možný jen pokud motor stojí, WH - zápis možný jen pokud jsou výstupy ve stavu vysoké impedance. Následuje popis nastavení registrů důležitých pro rychlé zprovoznění L6470.

Krokování

Pokud se točí motor, počet vykonaných kroků se zapisuje do registru ABS_POS. Ten má velikost 22 bitů a po přetečení se nuluje. Počet kroků odpovídá přednastavenému režimu krokování. V celokrokovém režimu jeden krok zvýší hodnotu registru o 1. Pokud je ale použit režim mikrokrokování 1/128, jeden celý krok bude odpovídat přírůstku 128.

Rychlostní profily

Pro nastavení rychlostních profilů slouží registry MAX_SPEED, MIN_SPEED, ACC a DEC. Všechny registry mají hodnoty vztaženy ke $\text{krok} \cdot \text{s}^{-1}$ a $\text{krok} \cdot \text{s}^{-2}$. Po jejich nastavení kontrolér nikdy nepřesáhne zadané hodnoty. Je třeba mít na paměti, že v těchto registrech se nenastavují přímo hodnoty v krocích za sekundu, ale v krocích na hodinový cyklus kontroléru. Ten je při použití interního oscilátoru 250 ns. Vzorce pro přepočty jsou v datasheetu v kapitole 9.1.

Nastavení napětí

Pokud motor ztrácí kroky, nebo se kontrolér zbytečně přehřívá, je nutné nastavit registry KVAL_HOLD, KVAL_RUN, KVAL_ACC a KVAL_DEC. Ty určují napětí přivedené na fáze motoru při zastavení, konstantní rychlosti, zrychlování a zpomalování. Tím se reguluje proud procházející vinutími. Napětí lze nastavit v rozsahu 0 až $0,996 V_S$ s rozlišením $0,004 V_S$, kde V_S je napájecí napětí výkonové části kontroléru. Je dobré nastavit nižší hodnoty pro zastavený motor a pro motor pohybující se konstantní rychlostí, především pokud mají vinutí motoru nízký odpor, sníží se tím ale krouticí moment.

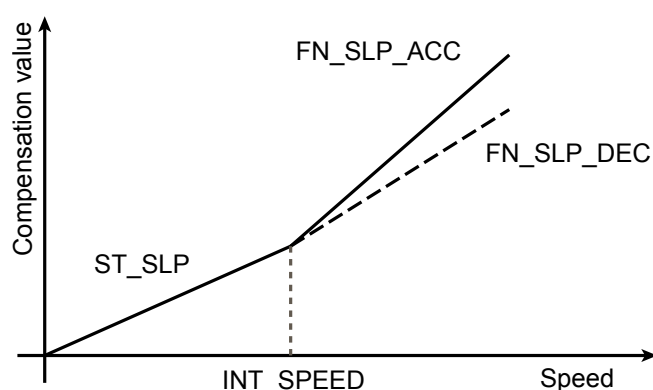


4.5 Kompenzace

Kontrolér L6470 je schopný provádět celou řadu kompenzačních opatření pro bezproblémový chod motoru. Kompenzace se provádí pomocí změny napětí na fázích motoru. Vztah pro výsledné napětí je uveden v podkapitole 4.5. Ve vztahu se uplatní parametry popsané níže.

Registr kompenzace zpětného elektromotorického napětí

Pro získání lepších parametrů motoru ve vyšších rychlostech se používá kompenzace zpětného elektromotorického napětí v cívkách, viz kapitola [2.5]. K tomu slouží úprava napětí na fázích KM v závislosti na rychlosti otáčení motoru. Úprava se provádí pomocí lomené přímky podle obrázku [8], jejíž sklon a místo zlomu se dá nastavit v registrech ST_SLP, FN_SLP_ACC, FN_SLP_DEC a INT_SPEED. Pro deaktivaci této kompenzace je třeba do všech registrů zapsat nuly.



Obrázek 8: Kompenzace BEMF. [6]

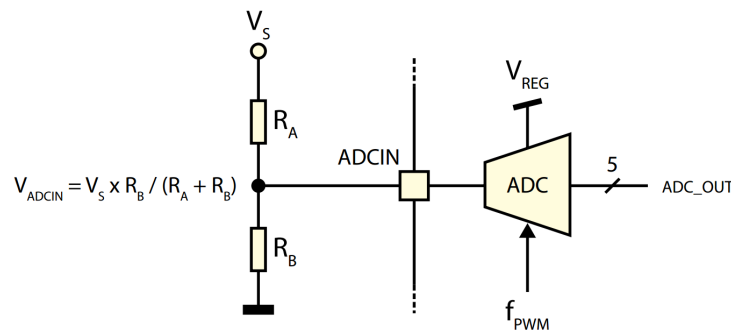
Teplotní kompenzace

Teplotní kompenzace redukuje změnu odporu vinutí motoru při vyšších teplotách. Pro její uplatnění je třeba teplotu motoru měřit, vyhodnotit a určit jakou hodnotu nastavíme do registru K_THERM. V defaultním nastavení je deaktivovaná. V modulu tato kompenzace není využita.



Kompenzace poklesu napájecího napětí

Tato kompenzace se využívá při náhlém poklesu napájecího napětí V_S . Hodnoty výstupu z PWM jsou totiž přímo úměrné V_S a při jeho poklesu by mohl být motor napájen nesprávně. Proto má L6470 jeden pin využít pro AD převodník, ke kterému se V_S připojí dle schématu na obrázku 9. Odpory R_A a R_B je třeba volit tak, aby $V_{ADCIN} = \frac{V_{REG}}{2}$, kde $V_{REG} = 3,3$ V je napětí interního regulátoru. Kompenzace se povolí ve status registru, viz kapitola 4.5.



Obrázek 9: Kompenzace poklesu V_S . [6]

Vyhodnocení kompenzací

Pro výsledné napětí V_{OUT} aplikované na fáze KM se používá vzorec (3), kde jsou zpracovány hodnoty všech kompenzací zmíněných výše. Hodnota *mikrostep* se mění v čase a udává poměrné zmenšení napětí na cívce při mikrokrokování. Je určena režimem kontroléru.

$$V_{OUT} = V_S \cdot (K_VAL + BEMF_COMP) \cdot VSCOMP \cdot K_THERM \cdot mikrostep \quad (3)$$

Na závěr je třeba dodat, že zmíněné kompenzace úzce souvisí s typem použitého motoru, hodnotou V_S , okolní teplotou, použitým mikrokrokováním a v jakém rychlostním rozsahu bude motor pracovat. Není zde proto uvedeno optimální nastavení těchto paramterů, nejlepší je odladit vše pro konkrétní aplikaci.

Status registr

Status registr slouží ke zjištění aktuálního stavu kontroléru. Pokud se např. nedaří vykonat s motorem nějaký pohyb, může se provést kontrola, zda není aktivní sedmý, či osmý bit



registru. To by ukazovalo na neproveditelný, nebo neznámý příkaz. Dále je zde uveden typ poruchy indikované pinem FLAG a jestli motor zrovna stojí, zrychluje, zpomaluje nebo se točí konstantní rychlostí. Pro podrobnější informace je doporučeno přejít kapitulu 9.2.22 v datasheetu.

Konfigurační registr

O konfiguračním registru jsou zmíněny jen nejdůležitější parametry. Důrazně se nedoporučuje nulovat sedmý bit registru, protože poté se nerozepnou můstky při přepětí. Hrozí poškození kontroléru. Také se zde nastavuje frekvence PWM. Při použití interního oscilátoru je frekvenční rozsah PWM od 2,8 MHz do 16 MHz. Dále je možné nastavit dobu přeběhu výstupu PWM od 180 V/ μ s do 530 V/ μ s. Pátý bit registru aktivuje kompenzaci poklesu napětí napájení kontroléru. Důležitý je čtvrtý bit registru, označovaný jako SW_MODE. Pokud je nastaven na hodnotu 0, uzemnění pinu SW způsobí zastavení KM. Toho bude využito k nastavení KM do výchozí polohy.

4.6 Stručný průvodce pro zprovoznění L6470

Pokud bude potřeba ovládat KM bez použití knihovny funkcí popsané v kapitole 6, je zde uveden seznam úkonů potřebných k roztočení motorku. Při zasílání dat do kontroléru je nutné dodržet formát dat popsaný v kapitole 4.3. Data jsou odesílána po jednotlivých bytech, MSB se odesílá jako první. První byte v příkazu určuje adresu příkazu, následují případné byty nesoucí samotná data. Všude jsou nastavovány maximální hodnoty, aby bylo patré, jak velké číslo je možné s příkazem odeslat. Jsou uvedena jen nezbytná nastavení pro bezpečný provoz kontroléru, roztočení a zastavení motoru:

- Nastavení max. napětí na vinutích. Druhý byte určuje hodnotu napětí viz 4.4. Opakovat i pro adresy 0x0A až 0x0C:
0x09 0xff
- Nastavení mikrokrokování na 1/128: **0x16 0x07**
- Nastavení maximální rychlosti na maximální hodnotu: **0x07 0x03 0xFF**
- Nastavení zrychlení na maximální hodnotu: **0x05 0x0F 0xFF**



- Nastavení zpomalení na maximální hodnotu: **0x05 0x0F 0xFF**
- Start motoru v kladném směru s maximální rychlostí dle nastaveného zrychlení: **0x51 0x0F 0xFF 0xFF**
- Plynulé zastavení motoru dle nastaveného zpomalení: **0x0B**



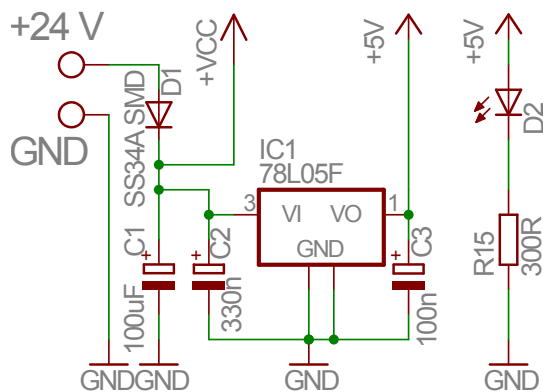
5 Schéma zapojení modulu

V této kapitole je popsáno schéma zapojení modulu. Jsou zde uvedeny výřezy částí schématu s popisem funkce. Úplné schéma je k nahlédnutí v příloze 27, včetně výstupu z návrhového systému Eagle a CAM souborů.

5.1 Napájení modulu

Na obrázku 10 je schéma napájení modulu. Modul je stavěn na napájení napětím 24 V. Maximální povolené napětí je 30 V, vyšší napětí by zničilo stabilizátor IC1. V případě použití jiného napájecího napětí je třeba změnit hodnoty odporů R3, R4, R5, R6, R7 a R8 podle pokynů v kapitole 4.5. Modul se napájí z externího zdroje, který není jeho součástí. Pokud je povolen nejvyšší možný proud na fázi KM, celý modul může mít odběr až 9 A, ve špičkách až 21 A. Na to je třeba při ovládání modulu pamatovat, zvolit vhodný napájecí zdroj a správně modul nastavit, viz kapitola 4.4. Při takto vysokých hodnotách proudu také nastává problém s chlazením modulu, viz kapitola 5.4.2

Napájecí napětí je do modulu přivedeno přes ochrannou diodu D1, která zabrání prepólování při prohození kontaktů. Blokovací kapacitor C1 pokrývá náhlé výkyvy napájecího napětí. K větvi s 24 V je připojen stabilizátor IC1, který napájí logickou část modulu. K identifikaci zapnutého napájení slouží LED dioda D2.



Obrázek 10: Schéma zapojení stabilizátoru.



5.3 Napojení na sběrnici SPI

Schéma napojení na sběrnici SPI je na obrázku 12. Všechny vstupní a výstupní signály jsou galvanicky odděleny od zbytku modulu pomocí dvou IO ADuM1401 a ADuM1400, viz [10]. Tím se předejde možnému rušení od výkonové části modulu. Signály jsou z modulu vedeny přes CON5.

Kromě obvyklých signálů používaných pro SPI jsou přidány dva signály FLAG a BUSY, viz kapitola 4.2. Výstupní piny udrží napětí při max. proudu 4 mA. Vstupní piny mají odběr maximálně 4 μ A. Pro napájení výstupní části modulu je na pinu 2 přivedeno napětí +5 V a na pin 1 zem. Maximální odběr pinem 2 je 15 mA při komunikační frekvenci 5 MHz. Použití 3,3 V logiky je možné, pokud na pin 2 přivedeme napětí 3,3 V. Všechny signály jsou kompatibilní s TTL logikou.

Galvanické oddělení není realizováno pomocí optronů, ale pomocí digitálních izolátorů ADuM od Analog Devices. Digitální izolátory jsou poměrně nové součástky. Signály se přenášejí pomocí elektromagnetické indukce mezi páry cívek. Toto řešení je výhodné díky nižší spotřebě, oproti optronům až desetkrát. Také reakční doba na změnu signálu je lepší.

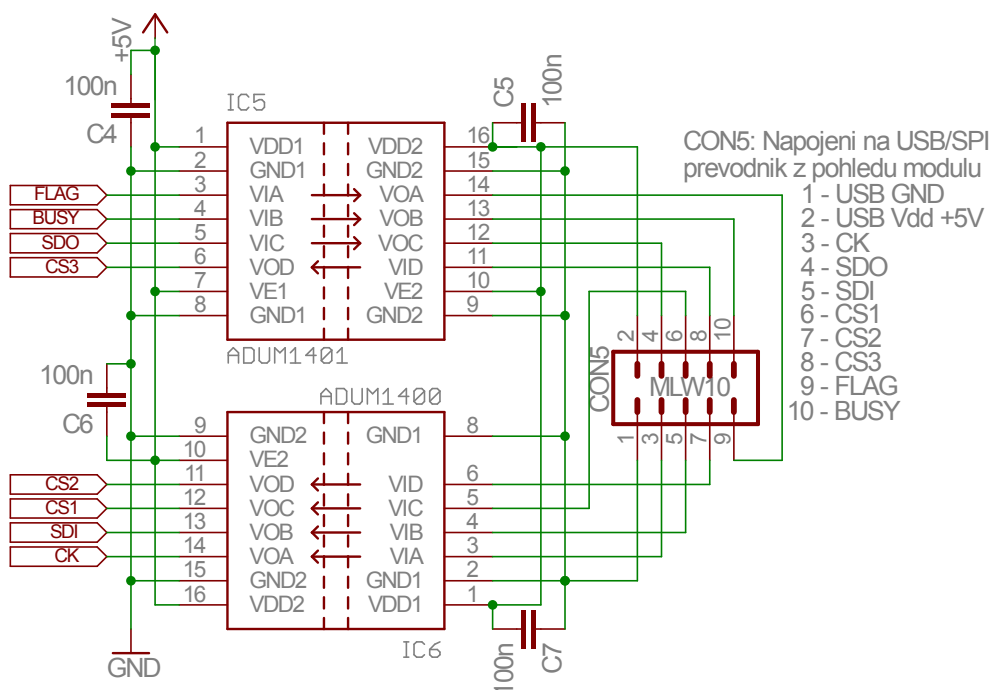
Pokud je CON5 odpojen a modul napájen, ADuM drží na pinech log. 1. Zabraňuje se tím náhodným stavům na sběrnici mezi ADuM a L6470. Toto platí i při vypnutém napájení modulu a připojeném konektoru CON5.

5.4 Deska plošných spojů

Tato podkapitola se zabývá konstrukcí DPS podle schématu uvedeném v příloze, viz obrázek 27. Z důvodu instalace do cytometru by DPS měla být co nejmenší. Na okraji desky se budou nacházet konektory pro napájení, výstupy ke KM, konektory pro koncové spínače a konektor pro napojení na SPI. Velmi důležité je navrhnout desku tak, aby šla vyrábět sériově. Výstupy práce je třeba poskytnout v elektronické podobě zpracovatelné firmami zabývajícími se výrobou DPS. Předpokládá se ruční osazování.

5.4.1 Konstrukce DPS

Návrh DPS byl proveden co nejjednodušeji, nebyly použity žádné drátové propoje. V případě potřeby se dá modul jednoduchou úpravou rozšířit o další kontroléry. Celkový pohled na



Obrázek 12: Schéma napojení modulu na SPI.

návrh DPS je v příloze, viz obrázek 23. Kromě celkového pohledu jsou v příloze i pohledy na obě strany desky a pohled ze strany součástek. Seznam součástek je uveden v tabulce 5 v příloze.

5.4.2 Popis DPS

Při konstrukci desky bylo nutné zohlednit poměrně velké proudy tekoucí vodivými cestami. Každý z kontrolérů může odebírat až 3 A r.m.s. Celkem tedy až 9 A. Tato hodnota je vzhledem k velikosti pouzdra kontroléru spíše teoretická, chlazení by bylo velmi náročné. Jsou předpokládány odběry maximálně 1 A na kontrolér, celkem tedy 3 A. Šířka výkonových cest byla zvolena na 2 mm. Dle převodní tabulky, viz [11], při použití mědi tlusté 16 μm a při ohřátí cesty nanejvýš o 20 °C tato šířka plně dostačuje. V návrhu jsou široké cesty napájení vedeny co nejbližší k pinům kontroléru. Stejně tak výstupy ke KM jsou řešeny tak, aby měly co největší šířku až k výstupním pinům. Také bylo pamatováno na kvalitní zemnění a na dostatečně široké zemní cesty ke konektoru napájení modulu, viz obrázky 24 a 25 v příloze.

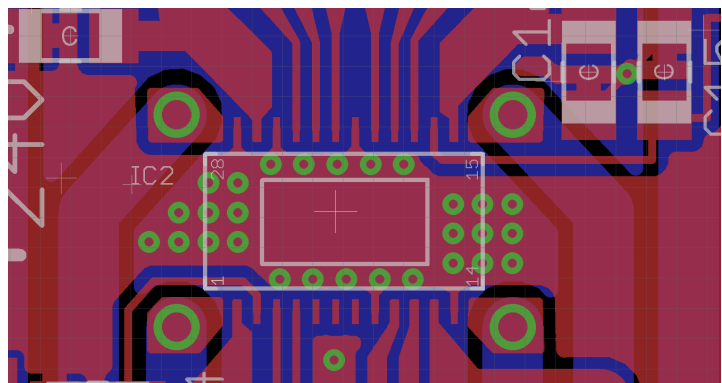
S velkými proudy tekoucími cestami jde ruku v ruce zahřívání kontrolérů. V datasheetu není ztrátový výkon uveden, přechodové odpory tranzistorů DMOS v sepnutém stavu jsou



pro horní stranu H-můstku $0,51 \, \Omega$ a pro dolní stranu $0,23 \, \Omega$. Je nutno podotknout, že toto platí pro teplotu PN přechodu $125 \, ^\circ\text{C}$, pro nižší teploty jsou hodnoty odporů menší. Také je třeba si uvědomit, že je použito PWM, takže tranzistory nejsou v sepnutém stavu po celou dobu. Předpokládejme proud $1 \, \text{A}$. Použitím ohmova zákona $P = I^2 R$ dostaneme ztrátový výkon $0,74 \, \text{W}$ pro jeden H-můstek. Celkem tedy $1,48 \, \text{W}$.

V datasheetu je uveden tepelný odpor mezi PN přechody a okolím IO $40 \, ^\circ\text{C}\cdot\text{W}^{-1}$. Tento údaj platí pro IO osazený na dvouvrstvé DPS s celkovou plochou mědi $12 \, \text{cm}^2$. Součinem ztrátového výkonu s tepelným odporem vyjde zvýšení teploty zhruba o $60 \, ^\circ\text{C}$ oproti teplotě okolí. Deska bude zabudovaná v cytometru, je třeba počítat s horším prouděním vzduchu. Proto je zvolena teplota okolí $40 \, ^\circ\text{C}$. Teplota kontroléru poté vychází na $100 \, ^\circ\text{C}$. Dle datasheetu je to teplota přijatelná pro provoz. Ale není vhodné, aby se modul takto zahříval. V návrhu DPS je proto třeba klást zvýšený důraz na správně provedené chladičí plochy mědi.

DPS je navržena tak, aby se kontroléry daly osadit malými pasivními chladiči. Detail chlazení je na obrázku 13. Kontrolér má chladičí plošku na spodní straně. Proto jsou kolem kontroléru umístěny prokovy odvádějící teplo na spodní stranu DPS. V praxi se poté ukázalo, že modul může pracovat i bez pasivních chladičů při proudech kolem $0,4 \, \text{A}$



Obrázek 13: Detail provedení chlazení kontroléru.

na kontrolér. Bylo zjištěno, že bude hodně záležet na režimu v jakém se budou KM pohybovat. Největší proudy tečou tranzistory při stojících KM. Proto je důležité nastavení maximálních proudů pomocí registrů KVAL, viz kapitola 4.4.

Vzhledem k tomu, že se cytometr bude používat v laboratořích, je třeba zabránit rušení způsobeným výskytem signálů vyšších kmitočtů. Proto je nutné z jedné strany DPS navrhnout rozlité plochy mědi. Ta bude zároveň sloužit k chlazení kontrolérů. K tomu je vyčleněna spodní strana desky, viz obrázek 25 v příloze. Na spodní straně desky se také nachází výkonové cesty ke KM a některé propoje, které nešly umístit na horní stranu DPS.



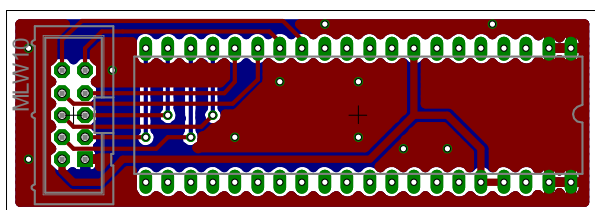
Kontrolér bude připojen k mikrokontroléru citlivému na rušení. Proto je provedeno galvanické oddělení komunikační části modulu. Nachází se v pravé části DPS. Zde bych chtěl také zmínit možnost jednoduchého rozšíření desky o další kontrolér. Stačí desku prodloužit a zkopírovat část s vodivými cestami kolem kontroléru. Pro zachování stejného počtu signálů CS na vstupní straně modulu by se musel použít demultiplexor. Jsou použity tři signály CS, teoreticky by tedy šlo připojit až sedm kontrolérů.

Při praktických testech bez použití pasivních chladičů se deska osvědčila. Kontroléry netrpí přehříváním do velikosti proudů zmíněných výše, nárazově je deska schopna uchladiť proudy v plném rozsahu možností kontrolérů. Komunikace s nadřazeným systémem funguje bez problémů i na nejvyšších frekvencích hodinového signálu. Během zkoušení desky byla zjištěna jediná vada. Při roztočení motorů na vysoké otáčky a rozepnutí H-můstek se přes ochranu diody dostává do napájecích obvodů napětí indukované zpomalujícími motory. Mohl by se prorazit pětivoltový stabilizátor, který je stavěn na vstupní napětí do 30 V. Tím by mohlo dojít ke zničení TTL logiky kontrolérů. Proto je třeba do další verze DPS přidat mezi napájecí cestu a zem zenerovu diodu v závěrném směru s $U_{zn} = 30 \text{ V}$. Současná verze modulu může takto bez problémů fungovat, protože pro řízení motorů není třeba H-můstky nikdy rozpojovat. Vše se děje po řízených rampách, které mohou mít i skokový průběh. V tom případě k tomuto jevu nedochází.

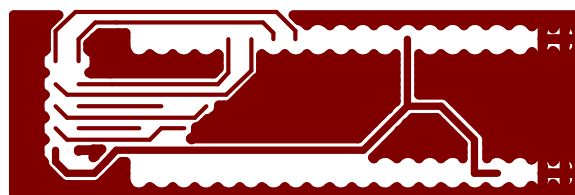
Výstupy ke KM jsou na desce označeny jako Chan 1 až Chan 3. V následujících kapitolách se předpokládá, že na Chan 1 se připojí X osa pohonu, osy Y a Z následují v abecedním pořadí. Fotky DPS jsou na obrázcích 28 a 29 v příloze.

5.4.3 Redukce pro SPI - USB konvertor s FT2232

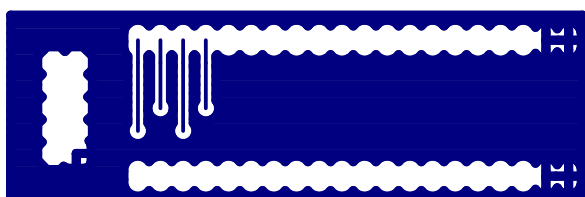
Pro odladění modulu bez použití mikrokontroléru byl použit komerčně dostupný konvertor USB - SPI s FT2232, viz [7]. K němu jsem si vyrobil redukci pro pohodlné připojení k modulu, viz obrázky 14, 24, 25 a 26. Konvertor se zasune do patice DIL20 napojené na konektor MLW10. Fotky redukce a konvertoru jsou na obrázcích 30 a 31 v příloze.



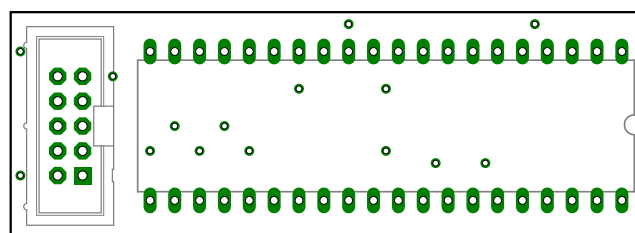
Obrázek 14: Celkový pohled na DPS redukce.



Obrázek 15: Pohled na horní stranu spojů.



Obrázek 16: Pohled na spodní stranu spojů.



Obrázek 17: Rozmístění součástek redukce.



6 Knihovna pro ovládání řídicího modulu

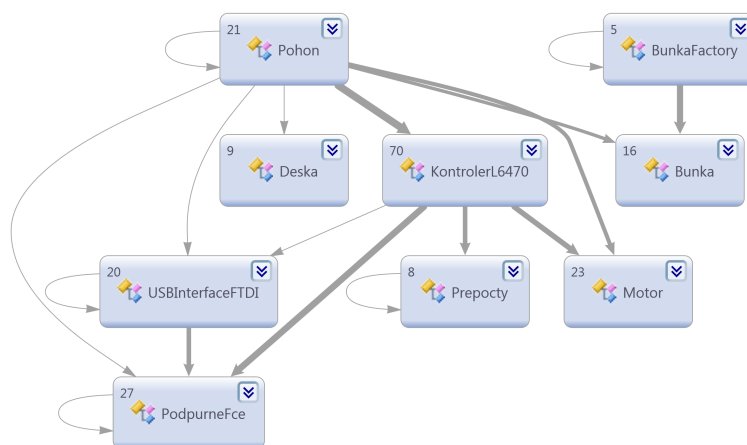
V této kapitole je popsán návrh dynamické knihovny (DLL) pro řízení modulu. DLL je vytvořena pro platformu .NET, zdrojové kódy jsou v jazyce C#. V principu celá DLL odráží fyzickou strukturu modulu včetně připojených krokových motorů a mechanismu os, které krokové motorky pohání.

Předpokládá se, že modul bude ovládán mikrokontrolérem uvnitř cytometru. Vzhledem k velké univerzálnosti modulu ale není možné vyloučit také použití modulu jako samostatného zařízení. V tom případě je třeba zajistit snadné napojení DLL na vrstvu, která zajistí přenos dat mezi nadřazeným systémem a samotným modulem. To je řešeno tak, že komunikace s nadřazeným systémem probíhá přes komunikační interface s pevně definovanými metodami. Ve zdrojových kódech je interface pojmenován jako `IUSBInterface`, viz tabulka 2 v příloze. V mém případě byl použit konvertor z USB na SPI od firmy FTDI. K němu výrobce poskytl ovladače a DLL s funkcemi pro jeho ovládání. Ovladače jsou na CD v příloze. Stačilo tedy vytvořit vrstvu mezi DLL výrobce a mojí DLL. Název této třídy je `USBInterfaceFTDI`. Takto si každý může vytvořit svou třídu pro ovládání vlastního komunikačního modulu.

Celková struktura DLL je znázorněna na obrázku 18. Vzhledem k rozsahu DLL nejsou popsány vnitřní funkce jednotlivých tříd. Namísto toho je uvedeno, jak DLL správně použít. Nejdůležitější třídy jsou `KontrolerL6470` a `USBInterfaceFTDI`.

Třída `KontrolerL6470` obsahuje sadu příkazů pro kontrolér, viz tabulka 2 v příloze, které jsou odesílány do modulu pomocí třídy `USBInterfaceFTDI`. Zasláné příkazy si třída `USBInterfaceFTDI` řadí do fronty podle pořadí, ve kterém je obdržela, ale neodešle je. Pro odeslání příkazů je třeba zavolat metodu `odesliData()`. Tím se výrazně sníží vytížení sběrnice USB. Zasílání dat po jednotlivých příkazech jde navíc proti principu sběrnice USB, která data posílá po částech s pevně danou velikostí. Zasílání příkazů po skupinách také zlepší souhru kontrolérů a eliminuje prodlevy mezi příkazy způsobené např. pomalým počítačem. Příkazy, které slouží pro načítání dat z registrů kontroléru a zjištění stavů pinů FLAG a BUSY, vrací hodnoty okamžitě.

Pro správnou funkci třídy `KontrolerL6470` je nezbytná třída `Motor`, kde jsou definovány parametry KM a jeho napojení na mechanismus, který kontrolér pohání. Do třídy `KontrolerL6470` se nezadávají hodnoty v krocích, ale v centimetrech. Pro převod na kroky



Obrázek 18: Diagram knihovny pro ovládání modulu.

se používá třída **Prepocety**. Do ní se posílá převáděná hodnota společně s parametry ze třídy **Motor**. To je velmi pohodlné, protože potom se není třeba zabývat tím, kolik kroků musí KM vykonat, aby se osa pohonu posunula o danou vzdálenost. Je třeba si uvědomit, že do výpočtu počtu kroků je třeba zahrnout převodový poměr na hřídeli KM, počet kroků na otáčku KM a nastavený mód mikrokrokování. V případě nastavení rychlosti a zrychlení je výpočet ještě složitější, protože hodnoty těchto registrů nejsou vztaženy ke krokům za sekundu, ale k cyklům kontroléru za sekundu, viz. kapitola 4.4. Třída **PodpurneFce** obsahuje řadu metod pro jednodušší práci s datovými typy a metody, které používám v kódu na více místech.

Třída **Bunka** si uchovává informaci o odběrové pozici, její polohu a pořadí. Třída **Deska** definuje rozměry mikrotitrační destičky a vzdálenost mezi odběrovými pozicemi.

Výše zmíněné třídy zastřešuje třída **Pohon**, kde je provedeno nastavení pro použití s modulem a obsahuje metody simulující funkci modulu jako pohonu odběrové jehly cytometru. Seznam metod je v tabulce 4 v příloze.

Celá struktura DLL je navržena tak, aby nebylo nutné použít pro ovládání modulu třídu **Pohon**. To umožňuje modul používat jako univerzální platformu pro pohon tří krokových motorů.



6.1 Příklad použití DLL

DLL pro ovládání řídicího modulu naleznete na CD v příloze pod názvem ModulKnihovna.dll.

V následujícím kódu v jazyce C# je uveden příklad jejího použití.

Vytvoření instance pro komunikaci s USB konvertorem a instancí kontrolérů:

```
//instance USB konvertoru
private IUSBInterface usbInterface = USBInterfaceFTDI.Instance;
//vytvoření instancí kontrolérů pro jednotlivé osy modulu
public KontrolerL6470 kontrolerX = new KontrolerL6470();
public KontrolerL6470 kontrolerY = new KontrolerL6470();
public KontrolerL6470 kontrolerZ = new KontrolerL6470();
```

Nastavení modulu:

```
usbInterface.pripojit(); // připojí USB konvertor
// nastavení kontroléru pro osu X, nutno provést pro každý kontrolér
kontrolerX.Jmeno = "KontrolerX";
// nastavení krokového motoru připojeného ke kontroléru
kontrolerX.motor = new Motor();
kontrolerX.motor.Jmeno = "MotorX";
kontrolerX.motor.pocetKrokuNaOtacku = 200;
// Nastavení průměru hnacího kola [cm]
kontrolerX.motor.prumerHnacihoKola = 3; // [cm]
kontrolerX.SetChipSelect = new byte[2] { 0x08, 0x00 }; // pozice pinu
    CS na sběrnici USB konvertoru FTDI. Pozici pinu určuje umístění bitu
    s hodnotou 1.
kontrolerY.SetChipSelect = new byte[2] { 0x10, 0x00 };
kontrolerZ.SetChipSelect = new byte[2] { 0x20, 0x00 };
```

Po úvodním nastavení již můžeme modulu začít posílat příkazy:

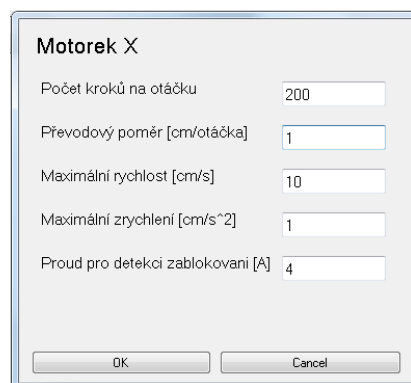
```
kontrolerX.setZrychleni(1); // nastaví zrychlení na 1 cm/s/s
kontrolerX.setZpomaleni(1); // nastaví zpomalení na 1 cm/s/s
kontrolerX.Run(true, 10); // motorek osy x se bude trvale otáčet
    rychlostí 10 cm/s po směru hod. ručiček
kontrolerY.pohniseOkroky(false, 128); // motorek osy y provede 128
    kroků proti směru hod. ručiček
kontrolerZ.GoHome(); // motorek osy z se vrátí do výchozí pozice
odesliPrikazy(); // až teď se příkazy najednou odešlou do modulu
kontrolerX.busy; // okamžitě vrátí stav pinu BUSY
kontrolerX.SoftHiz(); // plynule zastaví KM a rozepne H-můstky
usbInterface.odpoj() // odpojí USB konvertor
```



7 Demonstrační aplikace

Demonstrační aplikace je psaná v jazyce C# s použitím Microsoft Visual Studio, nachází se v příloze na přiloženém CD.

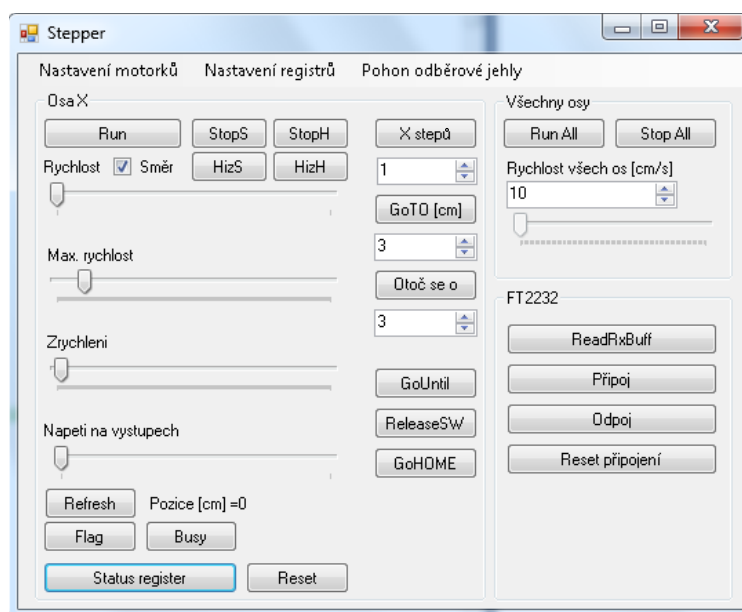
Aplikace se skládá z hlavního okna a tří podoken. Hlavní okno je na obrázku 20. Demonstruje základní funkce modulu na ose X (na desce označené jako Chan 1). V sekci „Osa X“ je možno nastavit maximální rychlost osy, aktuální rychlost, zrychlení, zpomalení a maximální napětí na cívkách KM. Dále je možno zjistit stav pinů Flag a Busy. Také můžeme vykonat určený počet kroků, přemístit se na určenou pozici a otočit se o vzálenost danou v cm. Tlačítka GoUntil, ReleaseSW a GoHOME demonstrují funkci koncového spínače. GoUntil provede pomalé najetí osy na koncový spínač a plynulé zastavení. ReleaseSW provede velmi pomalé odjetí z koncového spínače. Po rozepnutí spínače se motor okamžitě zastaví a nastaví se výchozí pozice osy (vynuluje se registr ABS_POS). Do této pozice je možné se kdykoliv vrátit pomocí tlačítka GoHOME. Tlačítkem Reset se resetuje kontrolér osy X. Tlačítka Refresh a Status slouží k načítání dat z kontroléru. V sekci „Všechny osy“ aplikace demonstruje pohyb všech os najednou.



Obrázek 19: Nastavení parametrů motoru.

Sekce „FT2232“ slouží pro ovládání konvertoru z SPI na USB. Tlačítko ReadRxBuff vypíše do konzole aktuální obsah vstupního registru konvertoru, funkce ostatních tlačítek je jasná z jejich názvu. Z horní lišty hlavního okna je možné zobrazit další podokna. Po kliknutí na položku „Nastavení motorků“ se zobrazí okno na obrázku 19. Zde je možné nastavit základní parametry motorku pro každou osu modulu.

V „Nastavení registrů“ se dají nastavit a zobrazit hodnoty v registrech kontrolérů jednotlivých os. Výběr konkrétního kontroléru se provede v rozbalovací nabídce a potvrdí se tlačítkem Select. Hodnoty jsou zobrazeny v hexadecimálním tvaru. Do kolonek lze zadat jen čísla v hexadecimálním tvaru a jen do velikosti podle délky registru, nekontroluje se ale správnost zadaných hodnot.



Obrázek 20: Hlavní okno demonstrační aplikace.

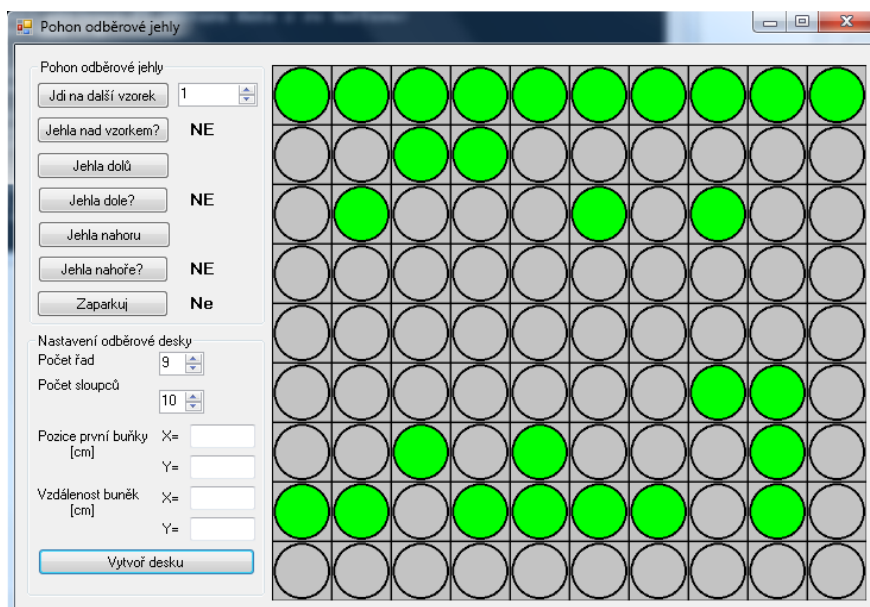
Poslední položka na vrchní liště zobrazí okno pro simulaci pohonu odběrové jehly v cytometru. Zde je možné vygenerovat odběrovou desku s libovolným počtem řad a sloupců a vzdálenostmi mezi odběrovými pozicemi. Ovládání modulu se děje přes třídu Pohon. Po volbě odběrových pozic pomocí myši se tlačítka v levé horní části ovládá sběr vzorků. Prvním tlačítkem se nastavuje pozice odběrové jehly nad zvolenou pozici, druhým tlačítkem se provede dotaz, zda už bylo pozice dosaženo. Dále následují tlačítka na ovládání posuvu odběrové jehly a kontrolu dosažení horní a dolní polohy jehly. Posledním tlačítkem se nastaví parkovací pozice odběrové jehly.

Nastavení registrů kontroléru

Refresh All Kontroler: X Select Close

Adresa [Hex]	Jmeno	Delka [b]	Aktualni hodnota [Hex]	Default [Hex]	Zapis/Cteni	Nastav	Refresh
0x01	ABS_POZ	22	0x000000	0x000000	R WS	Set	Refresh
0x02	EL_POZ	9	0x000	0x000	R WS	Set	Refresh
0x03	MARK	22	0x000000	0x000000	R W	Set	Refresh
0x04	SPEED	20	0x000000	0x000000	R	Set	Refresh
0x05	ACC	12	0x00a	0x00a	R WS	Set	Refresh
0x06	DEC	12	0x00a	0x00a	R WS	Set	Refresh
0x07	MAX_SPEED	10	0x041	0x041	R W	Set	Refresh
0x08	MIN_SPEED	13	0x0000	0x0000	R WS	Set	Refresh
0x15	FS_SPD	10	0x027	0x027	R W	Set	Refresh
0x09	KVAL_HOLD	8	0x29	0x29	R W	Set	Refresh
0x0A	KVAL_RUN	8	0x29	0x29	R W	Set	Refresh
0x0B	KVAL_ACC	8	0x29	0x29	R W	Set	Refresh
0x0C	KVAL_DEC	8	0x29	0x29	R W	Set	Refresh
0x0D	INT_SPD	14	0x0048	0x0048	R WH	Set	Refresh
0x0E	ST_SLP	8	0x19	0x19	R WH	Set	Refresh
0x0F	FN_SLP_ACC	8	0x29	0x29	R WH	Set	Refresh
0x10	FN_SLP_DEC	8	0x29	0x29	R WH	Set	Refresh
0x11	K_THERM	4	0x0	0x0	R W	Set	Refresh
0x12	ADC_OUT	5	0x19	0x00	R	Set	Refresh
0x13	OCD_TH	4	0x8	0x8	R W	Set	Refresh
0x14	STALL_TH	7	0x40	0x40	R W	Set	Refresh
0x16	STEP_MODE	8	0x07	0x07	R WH	Set	Refresh
0x17	ALARM_EN	8	0x7f	0xff	R WS	Set	Refresh
0x18	CONFIG	16	0x02e8	0x2e88	R WH	Set	Refresh
0x19	STATUS	16	0x07c3	0x0000	R	Set	Refresh

Obrázek 21: Registry kontroléru.



Obrázek 22: Demonstrace pohonu odběrové jehly.



Závěr

Podařilo se mi navrhnout a zrealizovat řízení pohonu odběrové jehly v průtokovém cytometru. Nejprve jsem se seznámil s principem odběrového mechanismu a požadavky na dynamiku motorů. Zvolil jsem řízení pomocí tří krokových motorů.

Následně jsem provedl výběr vhodného budiče krokového motoru. Pro pohon motorů jsem zvolil kontrolér L6470 od STMicroelectronics. Je to poměrně nový obvod s pokročilým komunikačním rozhraním. Poté jsem navrhl desku modulu, která obsahuje tři L6470 a rozhraní SPI. Návrh desky byl složitý, protože bylo třeba zajistit dostatečné chlazení kontrolérů a zároveň dodržet správnou šířku vodivých cest kvůli velkým proudům tekoucím kontroléry. Desku jsem si nechal vyrobit na zakázku, abych ověřil, zda ji bude možné vyrábět ve větším množství.

K desce jsem naprogramoval knihovnu funkcí pro snadné ovládání krokových motorů. Postupným vývojem jsem tak kromě funkcí pro řízení odběrové jehly v cytometru získal poměrně ucelený soubor příkazů, pomocí kterých se dá modul použít jako univerzální nástroj pro řízení krokových motorů. Velkou výhodou knihovny je to, že umožňuje zadávat vzdálenosti v centimetrech namísto v krocích a tak výrazně usnadňuje řízení motorků.

Nakonec jsem vytvořil demonstrační aplikaci, která ukazuje funkci celého pohonu. Aplikace také umožňuje nastavit veškeré parametry modulu pomocí editoru registrů kontroléru. Bohužel jsem neměl možnost odzkoušet modul na skutečném pohonu odběrové jehly, protože firma W&D, s kterou jsem spolupracoval, pohon nestačila dodat. Musel jsem si proto obstarat vlastní krokové motorky a vymyslet náhradní způsob prezentace modulu.

Další případnou práci na modulu vidím v možnosti vylepšit knihovnu funkcí pro ovládání modulu. Knihovna je napsaná v C# a pro zvýšení univerzálnosti by bylo dobré ji přepsat do jazyka C, nebo k ní napsat wrapper. Další prostor pro zlepšení by mohl být v rozšíření knihovny o další funkce. Zatím jsou osy pohonu schopny pohybu z bodu do bodu. Pro některé aplikace však může být požadována součinnost os taková, aby byly osy schopné vykreslit hladkou křivku. Také počítám s tím, že se může ve zkušebním provozu objevit nějaká drobná chyba.

Potěšilo mě, že moje práce bude použita nejen pro řízení pohonu odběrové jehly, ale také v projektu MPO TIP FR-TI1/591 - POCT, Vývoj moderního zařízení pro rychlou a



účinnou diagnostiku infekčních a geneticky podmíněných chorob člověka v režimu POCT, jehož nositelem je firma W&D a spoluřešitelem KTS a doc. Ivan Doležal, CSc. Modul bude řídit mechanismus vývojového vzorku pro pootáčení kartuší.



Literatura

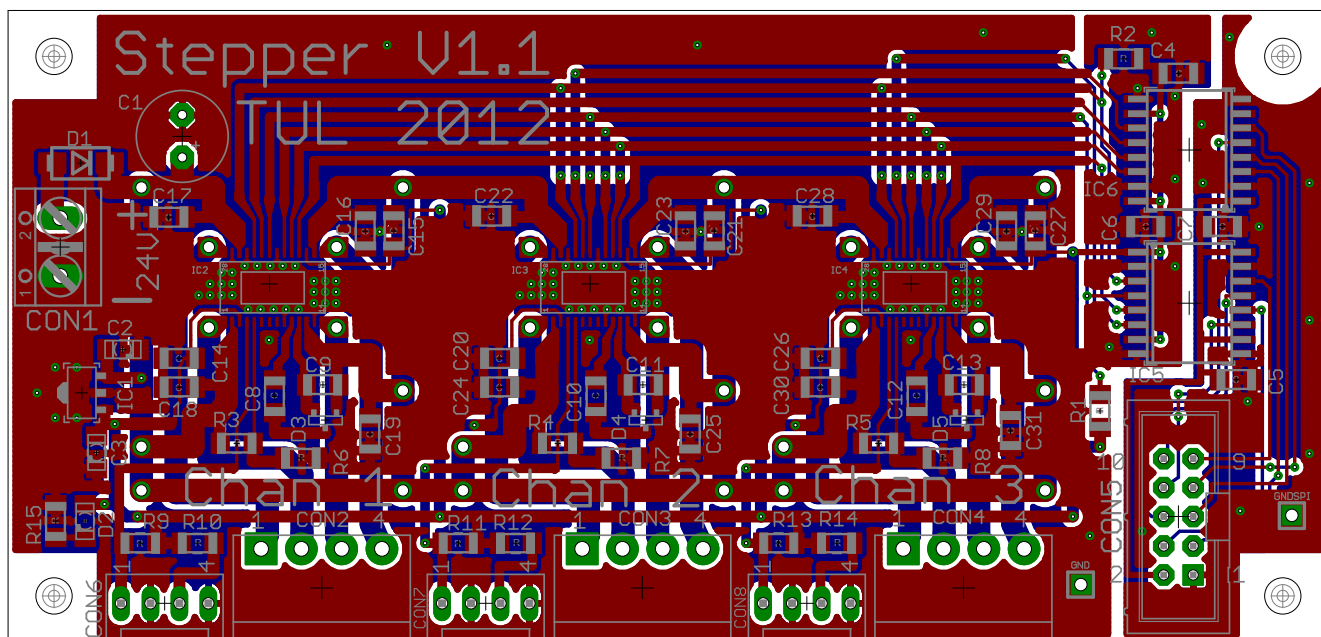
- [1] Rydlo, P; *Krokové motory a jejich řízení*. [Skriptum FM TUL], Liberec 2000. Dostupné z: <http://www.mti.tul.cz/files/ats/krok2.pdf>
- [2] Jones, D; *Control of Stepping Motors. A Tutorial*. The University of Iowa. Dostupné z: <http://www.divms.uiowa.edu/~jones/step>
- [3] Zhu J; *Electromechanical Systems. Stepping Motors and Their Power Electronic Drives*. University of Technology Sydney. Dostupné z: http://services.eng.uts.edu.au/cempe/subjects_JGZ/ems/ems_ch10_nt.pdf
- [4] <http://www.allegromicro.com/Products/Motor-Driver-And-Interface-ICs/Bipolar-Stepper-Motor-Drivers/A4980.aspx>
- [5] http://www.trinamic.com/index.php?option=com_content&view=article&id=176&Itemid=304
- [6] STMicroelectronics; *Kontrolér L6470*. [datasheet], 2011. Dostupné z: <http://www.st.com/internet/analog/product/248592.jsp>
- [7] MCU; *Modul s FTDI2232*. [datasheet], 2004. Dostupné z: <http://mcu.cz/news.php?item.467M>
- [8] David Swanson, Radek Stejskal; *Back-EMF method detects stepper-motor stall*, 2012. Dostupné z: http://www.edn.com/article/520450-Back_EMF_method_detects_stepper_motor_stall.php
- [9] Enika; *Jak vybrat krokový motor* [online], 2012. Dostupné z: http://www.enika.cz/download/Automatizac/Jak_vybrat_krokovy_motor.pdf
- [10] Analog Devices; *Quad-Channel Digital Isolators* [datasheet], 2012. Dostupné z: http://www.analog.com/static/imported-files/data_sheets/ADuM1400_1401_1402.pdf
- [11] Prago Board; *Tabulka proudové zatížitelnosti DPS* [tabulka], 2012. Dostupné z: http://www.pragoboard.cz/images/graf_spoj.jpg



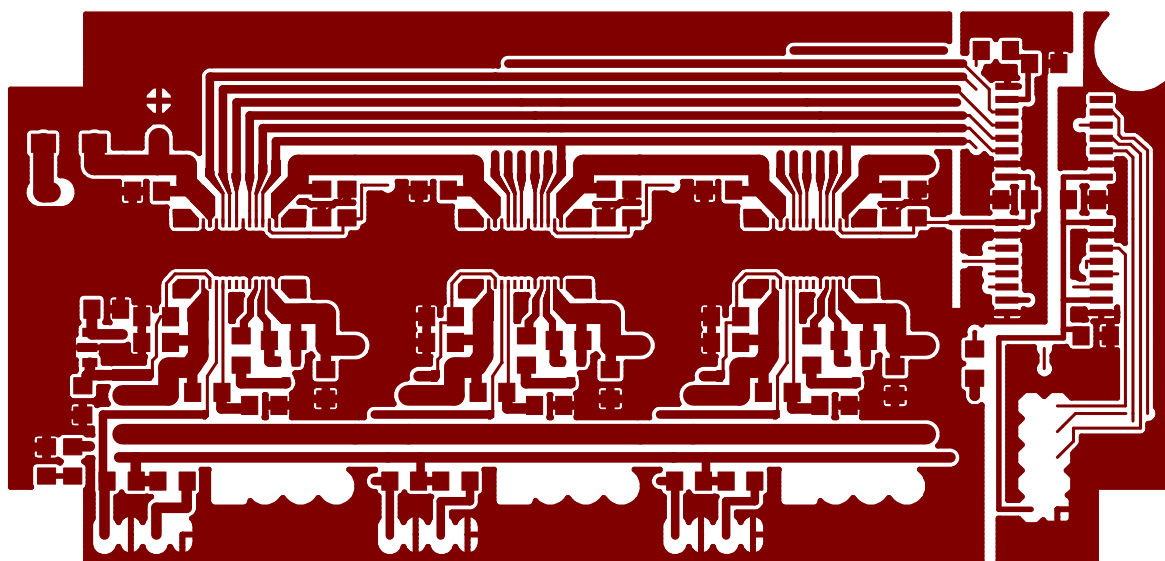
Přílohy

Přikládám potřebné obrázky a CD, které obsahuje následující složky:

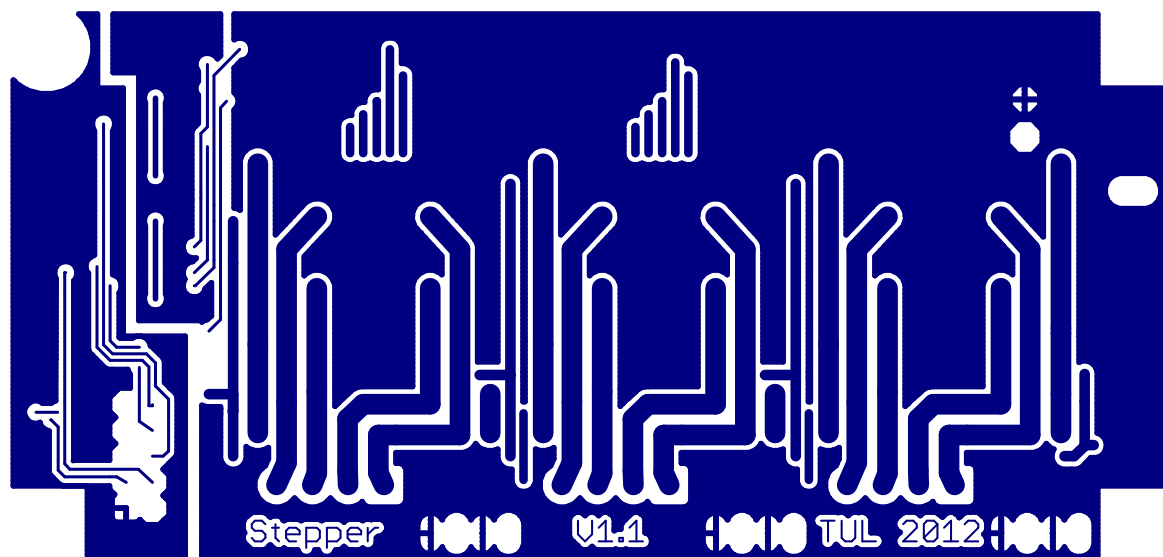
- soubor pdf s touto prací
- schémata navržená v programu Eagle
- demonstrační aplikaci a její projekt exportovaný z Microsoft Visual Studio
- DLL s funkcemi pro řízení modulu a její projekt exportovaný z Microsoft Visual Studio
- datasheety základních součástek



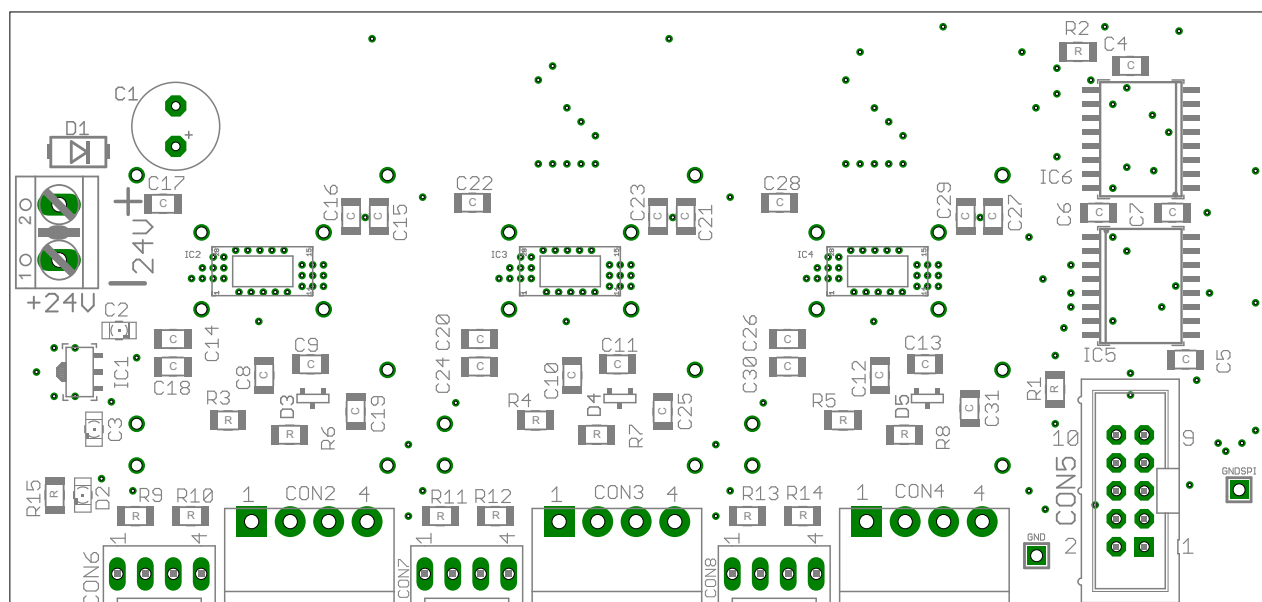
Obrázek 23: Celkový pohled na DPS modulu.



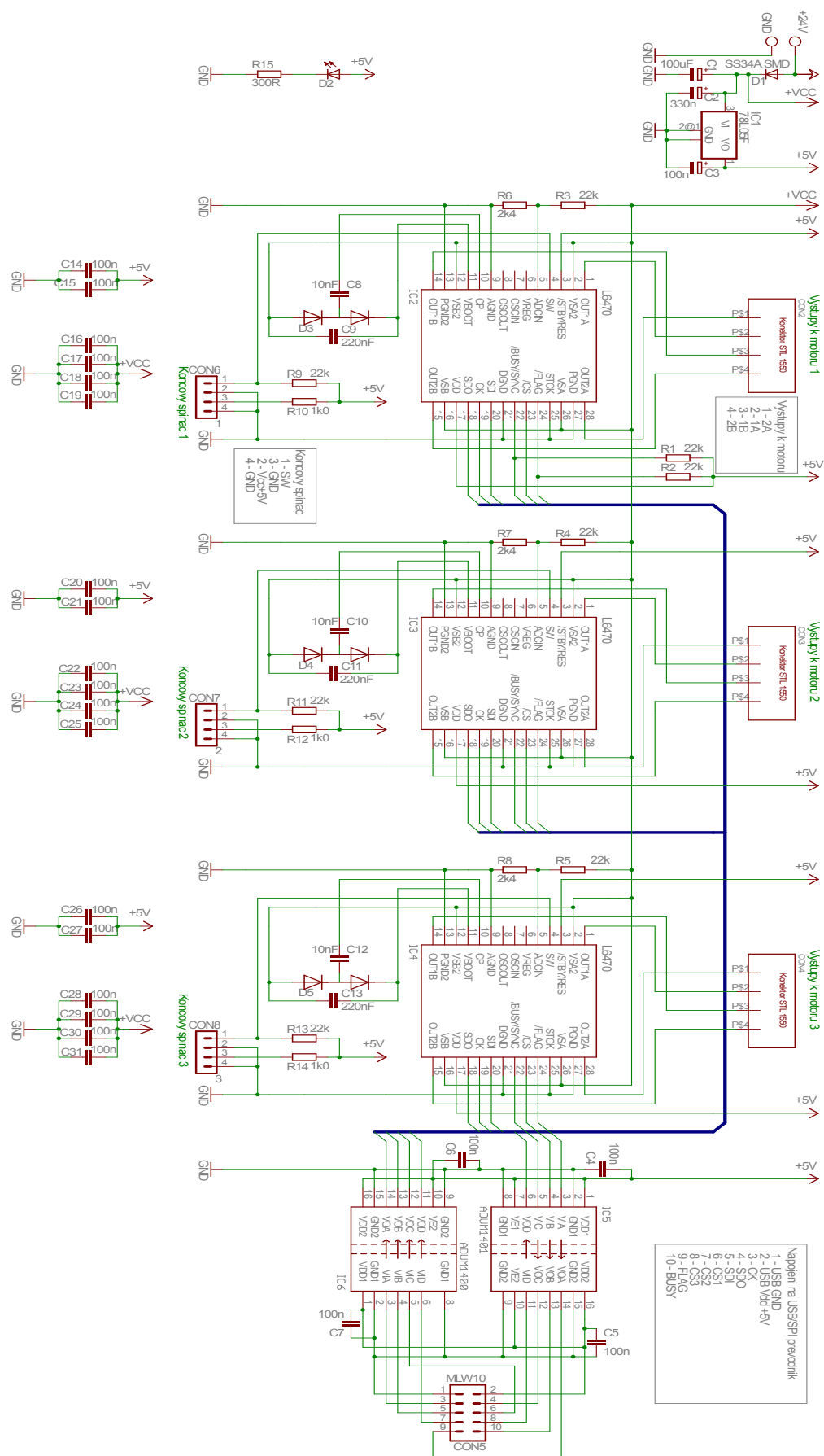
Obrázek 24: Pohled na horní stranu spojů DPS modulu.



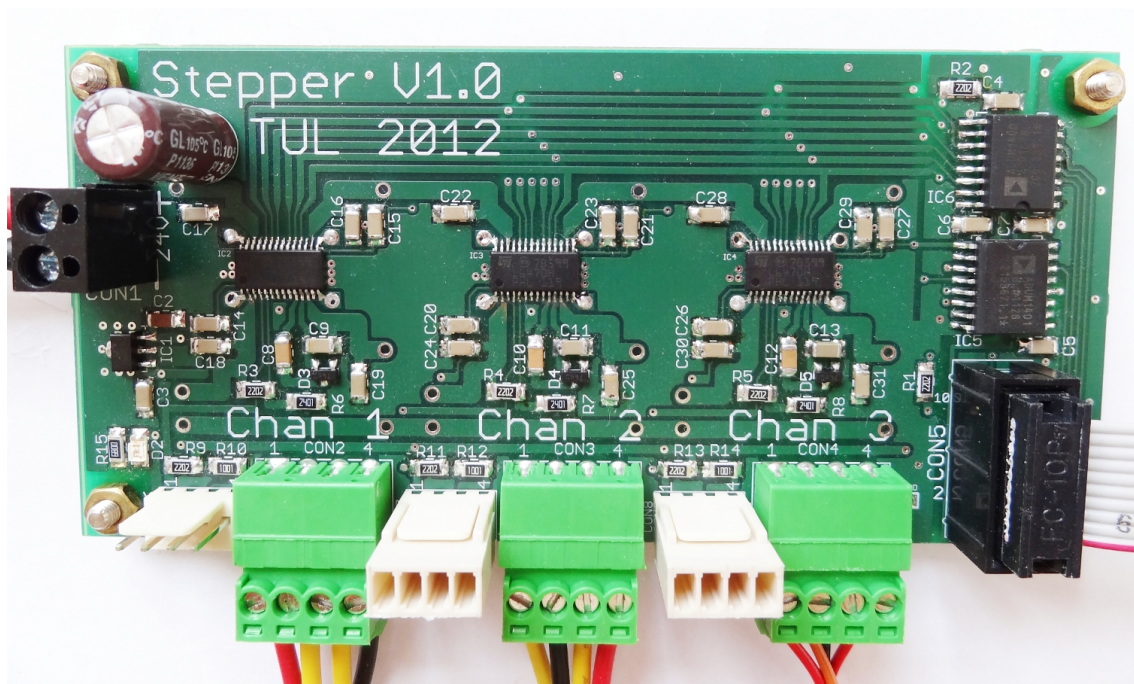
Obrázek 25: Pohled na spodní stranu spojů DPS modulu (zrcadleno).



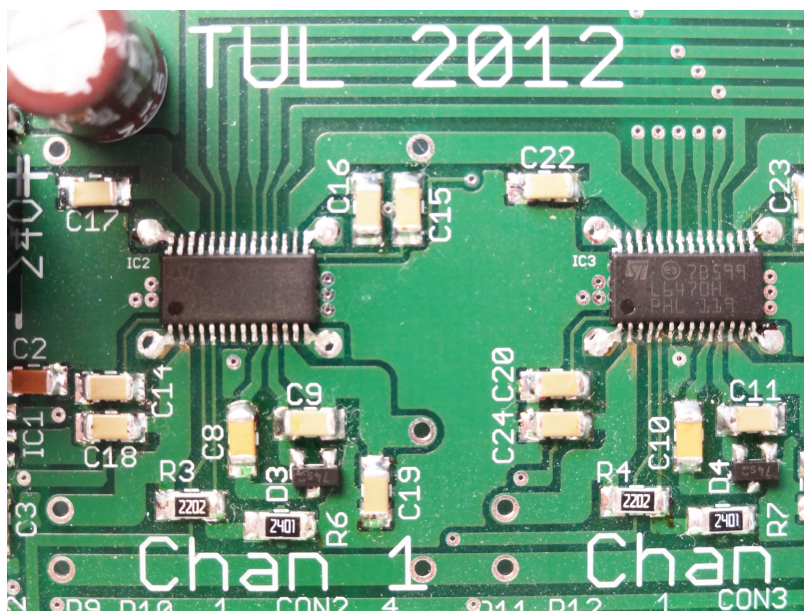
Obrázek 26: Pohled na stranu součástek DPS modulu.



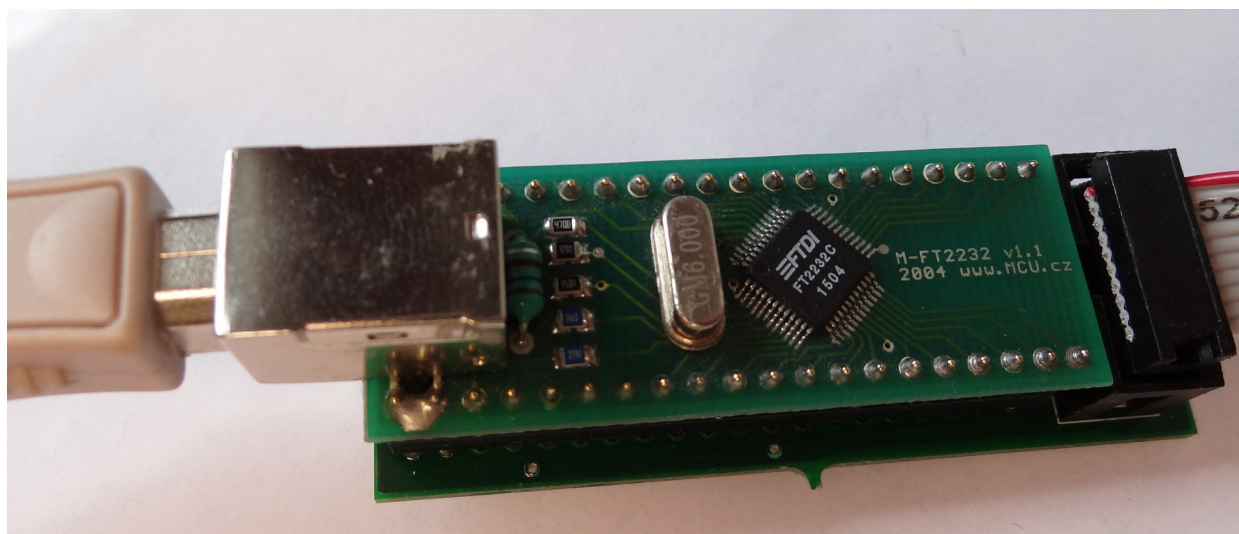
Obrázek 27: Úplné schéma zapojení modulu.



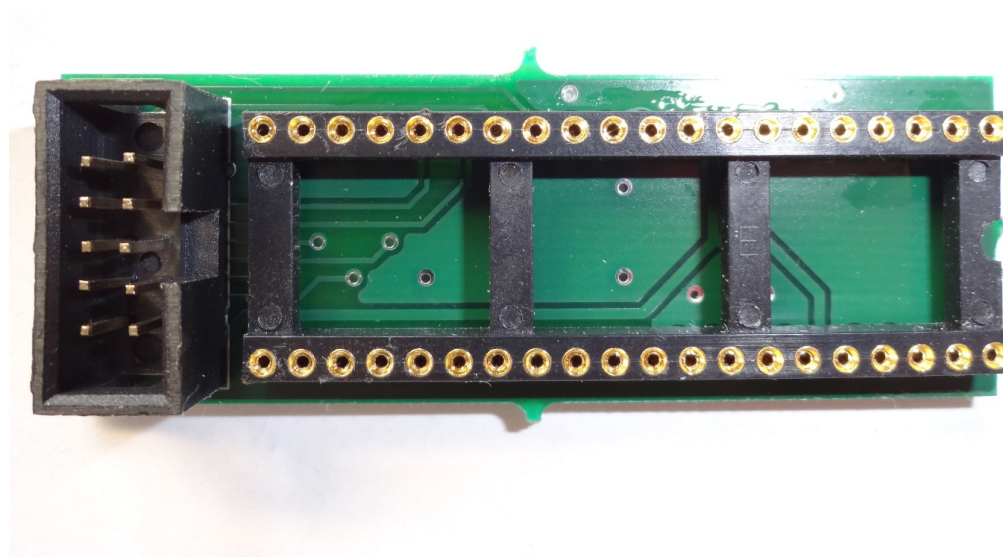
Obrázek 28: Fotka celku DPS.



Obrázek 29: Fotka detailu DPS.



Obrázek 30: Konvertor z USB na SPI.



Obrázek 31: Redukce pro konvertor z USB na SPI.



Tabulka 1: Třída KontrolerL6470.

Název metody	Popis
void Nop()	Prázdná instrukce.
void SetParam(byte param, byte[] data)	Nastaví parametr registru, param je adresa registru.
byte[] GetParam(byte param, int delka)	Načte parametr z registru.
void Run(bool smer, double rychlost)	Roztočí motorek danou rychlostí a směrem. smer = true je po směru hod. ručiček.
void Move(bool smer, double vzdalenost)	Posune osu v daném směru a vzdálenosti [cm].
void GoUntil(bool act, bool smerToceni, double speed)	Posouvá osu v daném směru a rychlostí [cm/s], dokud nesezne koncový spínač.
void odesliData()	Odešle frontu s daty.
void GoHome()	Návrat na výchozí pozici.
void GoMark()	Návrat na Mark pozici.
void ResetPos()	Resetuje výchozí pozici.
void ResetDevice()	Resetuje kontrolér.
void SoftStop()	Provede plynulé zastavení dle registru ACC.
void HardStop()	Provede okamžité zastavení.
void SoftHiz()	Provede plynulé zastavení dle registru ACC a rozpojí H-můstky.
void HardHiz()	Rozpojí H-můstky.
byte[] GetStatus()	Navrátí hodnotu status registru.
void pohniseOkroky(bool smer, int kroky)	Otočí motorkem o daný počet kroků.
void setMaxRychlost(double maxRychlost)	Nastaví max. rychlost osy [cm/s]



Tabulka 2: Interface IUSBInterface.

Název metody	Popis
<code>void pripojit();</code>	Aktivuje komunikační modul.
<code>bool nastavSPI();</code>	Provede nastavení sběrnice SPI.
<code>string odpoj();</code>	Odpojí komunikační modul.
<code>string reset();</code>	Resetuje komunikační modul.
<code>string SPInastavRychlostCLK(int delic);</code>	Nastaví rychlost sběrnice SPI.
<code>string dataKposlani(byte[] data,byte[] cs);</code>	Zařadí data k odeslání do fronty, MSB nejdříve, cs značí pozici chipselectu.
<code>void odesliData();</code>	Odešle frontu s daty.
<code>byte[] SPIdataOboji(byte data, int delkaPriijatych, byte[] chipSel);</code>	Odešle jede byte a přijme n bytů dle int delkaprijatych. chipSel značí pozici chipselectu.
<code>byte pinyLowNacti();</code>	Načte stav prvních 8 pinů sběrnice komunikačního zařízení.
<code>byte pinyHighNacti();</code>	Načte posledních 8 pinů sběrnice komunikačního zařízení.
<code>byte[] pinyNacti();</code>	Načte stav všech pinů sběrnice komunikačního zařízení. Hodnoty vrací v poli bytů o velikosti 2.



Tabulka 3: Třída KontrolerL6470 pokračování.

Název metody	Popis
setZrychleni(double zrychleni)	Nastaví zrychlení osy [$\text{cm}\cdot\text{s}^{-2}$]
setZpomaleni(double zpomaleni)	Nastaví zpomalení osy [$\text{cm}\cdot\text{s}^{-2}$]
bool getSteplosA	Detekuje ztrátu kroku na fázi A.
bool getSteplosB	Detekuje ztrátu kroku na fázi B.
void aretace()	Provede aretaci osy. Výsledek v bool aretaceOK.
void rychlostAretace(double rAretace)	Nastaví rychlost aretace osy.

Tabulka 4: Třída Pohon.

Název metody	Popis
void inicializuj()	Slouží k inicializaci modulu.
bool jdiNaDalsiVzorek()	Přejde na další vzorek. Neúspěch = false.
bool naVzorku()	Dotaz, zda je modul nad vzorkem.
int zbyvaVzorku	Vrací počet zbývajících vzorků.
double aktualniPoziceX	Vrací aktuální pozici osy X.
double aktualniPoziceY	Vrací aktuální pozici osy Y.
double aktualniPoziceZ	Vrací aktuální pozici osy Z.
bool vProvozu	Detekuje, zda je modul v provozu.
Deska deska	Proměnná! Vrací a nastavuje objekt Deska.
bool jehlaDolu()	Posune odběrovou jehlu dolů.
bool jehlaNahoru()	Posune odběrovou jehlu nahoru.
bool jehlaDole()	Dotaz, zda je jehla dole.
bool jehlaNahore()	Dotaz, zda je jehla nahore.



Tabulka 5: Seznam součástek.

Značení	Hodnota	Popis	Pouzdro
C1	100uF	elektrolyt	RM3,5
C2	330n	keramika	1206
C3 až C8	100n	keramika	1206
C9	220n	keramika	1206
C10	10n	keramika	1206
C11	220n	keramika	1206
C12	10n	keramika	1206
C13 až C31	100n	keramika	1206
R1 až R9	22k	rezistor	1206
R10	1k0	rezistor	1206
R11	22k	rezistor	1206
R12	1k0	rezistor	1206
R13	22k	rezistor	1206
R14	1k0	rezistor	1206
R15	300R	rezistor	1206
D1		M7	DO21BA
D2		LED zelená	1206
D1 až D5		BAV99	SOT23
IC1		stabilizátor	78L05F
IC2 až IC4	HTSSOP28	STM L6470	
IC5		ADUM1400	SO16W
IC6		ADUM1402	SO16W
CON1		PTR AK500/2DS	
CON2 až CON4		PTR STL1550/4G	
CON5		MLW10GT	
CON6 až CON8		PSH02-04P	